UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

Desagregação de consumos de *smart homes* por tipologia

Alexandre Pereira Martins Rafael Torrejano

Mestrado em Estatística e Investigação Operacional
Especialização em Estatística

Relatório de Estágio orientado por:
Prof. Doutor Fernando José Araújo Correia da Ponte Sequeira
Eng. Pedro Magalhães Adão

2018

Agradecimentos

Dedico este trabalho aos meus pais pelo apoio incondicional que me deram durante todo o meu percurso, não só na faculdade mas ao longo de toda a minha vida. Sem os seus ensinamentos não seria detentor da vontade e dedicação que este trabalho exigiu. Agradeço também à Beatriz, a melhor irmã que alguém pode pedir.

Agradeço imenso à Professora Helena Iglésias Pereira, que foi minha orientadora até à fase final da escrita do relatório. Enquanto lhe foi possível, foi uma ajuda preciosa e incansável no desenvolvimento do trabalho, e por isto estou-lhe eternamente grato.

Agradeço ao Pedro Adão pela partilha da sua sabedoria pragmática e pelas soluções que as suas sugestões excelentes me permitiram alcançar. Agradeço ao Professor Fernando Sequeira, por ter assumido a minha orientação na fase final e pela sua boa disposição que me encorajou a encarar os desafios finais com calma e tranquilidade.

Agradeço ao Dinis, ao Pedro, à avó Elisa e aos meus colegas e amigos Jorge, Miguel, Mariana, Estêvão, Gonçalo, Catarina e Alexandra pelo seu apoio e amizade que não se resumiram à sua presença no dia da defesa, mas sim a todo o percurso.

Há muitos nomes de amigos que não menciono, no entanto eles sabem quem são e quão importante a sua amizade é para mim.

Por último, mas muito importante, agradeço à Sara, minha companheira, meu lugar seguro. A sua presença tranquilizante ofuscou os problemas com que me deparei ao longo deste caminho.

Resumo

O presente trabalho tem como tema a desagregação de consumos no setor doméstico e a sua aplicação ao projeto *re:dy* da EDP, cuja frequência de amostragem dos valores de consumo dos clientes é relativamente baixa. Os conceitos da desagregação de consumos e do projeto EDP *re:dy* são explicados no capítulo de contextualização. O capítulo de metodologia contém uma explicação para cada um dos métodos matemáticos que foram utilizados proeminentemente durante o estágio. As abordagens exploradas para resolução do problema podem ser essencialmente divididas em dois processos preditivos: o processo utilizado para prever o consumo de frigoríficos e máquinas e o processo desenvolvido especificamente para a estimação do consumo de aquecimento ambiente.

O primeiro processo segue uma estrutura de *Ensemble Learning* contando com 7 algoritmos e 5 meta-algoritmos cujos desempenhos são comparados após a análise dos valores preditos para o consumo de frigoríficos e máquinas dos clientes. Antes da construção do processo, as amostras dos consumos das categorias de equipamento em questão foram sujeitas a uma análise exploratória para facilitar a escolha de algoritmos mais adequados. O conjunto de variáveis independentes (*input* do algoritmo) foi derivado da informação disponível para todos os clientes e processado através de uma análise em componentes principais.

O processo preditivo para consumo de aquecimento ambiente foi desenvolvido estudando o impacto desta classe de equipamentos no consumo global dos clientes ao longo do ano. Ao contrário do primeiro processo, que é maioritariamente constituído por modelos estatísticos, este é um algoritmo empírico.

Por fim, no capítulo de discussão, analisam-se as várias abordagens e possíveis direções futuras da sua aplicação ao projeto.

Palavras-chave: Desagregação de consumos; baixa frequência; EDP *re:dy*; *Ensemble Learning*; *Data Science*.

Abstract

The present work is focused on household energy disaggregation and its application to EDP's re:dy project, that has a relatively low sampling frequency for clients' consumption values. The concepts of energy disaggregation and of the re:dy project are explained in a contextualization chapter. The methodology chapter contains explanations for each mathematical method that was prominently used throughout the internship. The approaches to the problem can be split into two predictive processes: the process used to predict the consumption of fridges and washers and the process specifically designed to estimate the consumption of heaters.

The first process follows an Ensemble Learning framework, including 7 algorithms and 5 meta algorithms whose performances are compared after analyzing the predicted values for fridges' and washers' consumption. Before the construction of the predictive process, the consumptions of the said equipment categories were analyzed to help choose better suited algorithms. The set of input variables was derived from the information available for every client and processed through a principal component analysis.

The predictive process for heater consumption was developed by studying the impact of this equipment class on the global consumption of the clients throughout the year. Unlike the first approach, that was mainly composed of statistical models, this one is an empirical algorithm.

Finally, in the discussion chapter, both approaches are analyzed as well as the future of their application to the project.

Keywords: Energy Disaggregation; low frequency; EDP re:dy; Ensemble Learning; Data Science.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de siglas

DDSC – Discriminative Disaggregation Sparse Coding
EDP – Energias de Portugal
kWh – QuiloWatt-hora
MAE – Mean Absolute Error (erro absoluto médio)
NILM – Nonintrusive Load Monitoring
re:dy – Remote Energy Dynamics
RMSE – Root Mean Squared Error (raiz do erro quadrático médio)

1. Contextualização

1.1 Projeto EDP *re:dy*

O EDP *re:dy* (*remote energy dynamics*) é um projeto que permite aos seus clientes monitorizar e controlar os equipamentos elétricos das suas casas através de uma aplicação para *smartphones*.

Figura 1.1: Ilustração de apresentação da aplicação para *smartphones* "EDP *re:dy smart home*".

Entre as funcionalidades do EDP *re:dy* estão: desligar e ligar equipamentos elétricos, programá-los para funcionamento a uma certa hora do dia, controlar o seu funcionamento e monitorizar o seu consumo (alertando o cliente quando são detetadas situações de consumo excessivo), tudo remotamente através do telemóvel pessoal do cliente.

A funcionalidade sobre a qual o estágio incide é a monitorização do consumo elétrico global da casa e de aparelhos elétricos isolados. Esta monitorização é efetuada com base em medições de consumo elétrico (em kWh), que são processadas de 15 em 15 minutos e cujos valores são recolhidos para análise e apresentação ao cliente.

As medições do consumo de aparelhos isolados são efetuadas por *plugs/meters* – dispositivos que medem a energia elétrica transferida da tomada elétrica para o aparelho em questão. O consumo total da casa é monitorizado por um *smartmeter* que substitui o contador de eletricidade clássico.

A EDP *re:dy box* é o dispositivo responsável por receber toda a informação. Esta recebe:

- informação das *plugs/meters* por comunicação sem fios;
- informação do *smartmeter* (que mede o consumo total da casa) através de uma ligação por cabo de eletricidade.

A EDP *re:dy box* processa toda esta informação, enviando-a pela *Internet* para uma base de dados que contém dados de consumo de todos os clientes *re:dy*.

Figura 1.2: À Esquerda: EDP *re:dy plugs*; À direita: EDP *re:dy box*.

1.2 Desagregação de consumos

Desagregação de consumos de energia elétrica é a prática de estimar o consumo de cada aparelho elétrico ou de cada classe de aparelhos elétricos (consumos parciais) de uma casa ou edifício, partindo da informação sobre o seu consumo total (agregado).

O *feedback* relativo aos consumos parciais de um indivíduo pode ter um impacto significativo no seu comportamento, reduzindo o consumo total de energia elétrica [1]. No entanto, a monitorização de todos ou da maioria dos consumos parciais de uma habitação é bastante dispendiosa, o que leva à necessidade da existência de algoritmos que estimem os consumos parciais com base em informação mais amplamente disponível.

Os algoritmos desenvolvidos para o efeito de desagregação de consumo são frequentemente chamados *NILM (Nonintrusive Load Monitoring)*, uma vez que não requerem medições aos aparelhos elétricos isolados da casa para obter uma repartição do seu consumo global por aparelho ou tipologia. Por vezes, estas técnicas são também designadas de *NIALM* ou *NALM (Nonintrusive Appliance Load Monitoring)*.

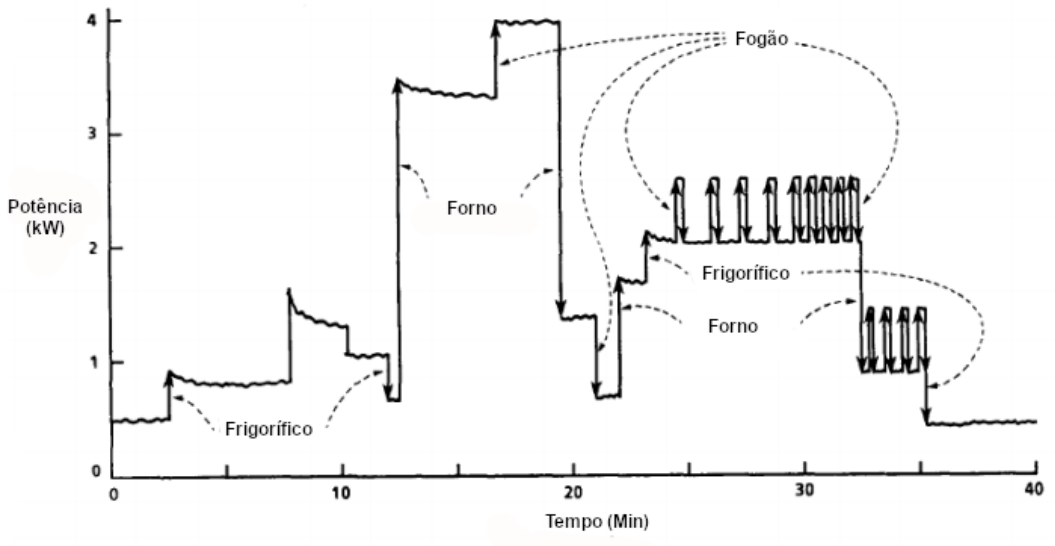


Figura 1.3: Assinatura elétrica de alta frequência com padrões de consumo identificados [2].

Os algoritmos *NILM* são, geralmente, desenvolvidos para casos de estudo específicos, de acordo com a informação sobre o consumo elétrico que está disponível.

Quando a recolha da informação do consumo total da casa ou edifício é efetuada com frequência alta ($\gg$ 1Hz), geralmente, os algoritmos são desenvolvidos em volta da capacidade de deteção de mudanças de estado e identificação dos aparelhos elétricos que causaram as ditas alterações na série temporal do consumo agregado (também designada de assinatura elétrica) [3] (Figura 1.3).

Os padrões de consumo dos vários tipos de aparelhos elétricos tornam-se mais difíceis de diferenciar a partir da assinatura elétrica global da habitação quando se consideram frequências de amostragem mais baixas. O tema da frequência de amostragem é uma preocupação recorrente na comunidade científica [4, 5], uma vez que, no geral, as empresas que implementam serviços de *smart metering* utilizam medidores que apenas permitem intervalos entre medições na ordem dos minutos ou das horas. Nestes casos, a previsão dos consumos parciais pode ter como base a análise do comportamento do consumo global do indivíduo (ao longo do dia, semana, ano, etc.), características estáticas do indivíduo e outras variáveis que se possam relacionar com o consumo elétrico (Exemplo: dados sobre o clima na zona da casa/edifício em questão).

A maioria das metodologias de desagregação de consumos são construídas com a aplicação ao consumo de energia elétrica. No entanto, existe potencial de aplicação deste tipo de técnicas a outros tipos de consumo como consumo de energia global (não só elétrica) ou consumo de água [6].

1.3 Objetivo do estágio

A funcionalidade de monitorização de consumos do EDP *re:dy*, pode ser uma grande vantagem em termos de poupança de energia para os clientes. No entanto a grande maioria dos clientes não possui *plugs* suficientes para obter uma boa descrição do consumo da sua casa. Estes clientes beneficiam menos da funcionalidade mencionada por receberem informação menos completa sobre os seus consumos parciais (Figura 1.3).

Figura 1.4: Gráfico circular referente à divisão mensal de consumos de um cliente.

O objetivo do estágio é a construção de um algoritmo capaz de estimar consumos parciais desconhecidos partindo da informação disponível. Desta forma seria possível apresentar um relatório mais detalhado aos clientes e alertá-los relativamente às possíveis causas do seu excesso de consumo. Idealmente, utilizando esta informação, os clientes serão capazes de reduzir o seu consumo mesmo possuindo poucas *plugs*.

Uma funcionalidade recente do EDP *re:dy* permite ao cliente listar os seus aparelhos elétricos através do preenchimento de um questionário na aplicação móvel. Embora as respostas a estes questionário ainda não sejam em grande número (aquando de julho de 2018), esta funcionalidade

reduz a necessidade de que o algoritmo tenha a capacidade de identificar aparelhos sem qualquer informação prévia sobre a lista de aparelhos do cliente. O objetivo do algoritmo é apenas estimar consumos parciais para os quais se sabe que existem na habitação do cliente em questão (embora tenham valor desconhecido).

1.4 Revisão de literatura

O desenvolvimento de metodologias de desagregação de consumos de energia começou em 1984 com George Hart [7]. Hart desenvolveu metodologias de identificação de aparelhos através da identificação de padrões e mudanças de estado na assinatura elétrica da casa. Uma vez que a soma dos consumos dos aparelhos é o consumo total da casa, a identificação do conjunto de aparelhos ligados a cada intervalo de tempo foi formulada como um problema de otimização combinatória. Além de dar um início da área de estudo, Hart foi o maior impulsionador das metodologias de desagregação de energia nas primeiras décadas, sendo autor de 18 publicações relacionadas com o tema até 1995. Até 2010, as publicações da área concentram-se no desenvolvimento de algoritmos que identificam os aparelhos através da assinatura elétrica de frequência elevada.

Com o início da comercialização de *smartmeters* na década de 2001-2010, surgiu a preocupação de criar métodos capazes de identificar corretamente os consumos de cada aparelho elétrico com frequências de amostragem do consumo global mais baixas. A primeira metodologia capaz de desagregação de consumo através de dados recolhidos com minutos de intervalo surge em 2010 com Kolter, Batra e Ng [4]. Esta metodologia, designada *Discriminative Disaggregation Sparse Coding* (*DDSC*) pelos autores, identifica padrões semanais nas séries temporais, considerando a hora de consumo um fator importante para a identificação do aparelho elétrico em utilização. O algoritmo é uma aplicação de um método de funções base e dicionário, um tipo de modelo que consiste na decomposição de uma matriz de dados em uma matriz de ativações (matriz de codificação) e uma matriz de funções base (matriz dicionário) [8]. Neste caso, o conjunto de treino inclui uma matriz de dados para cada categoria de equipamento elétrico considerada, em que cada coluna é uma semana de observações. A matriz de ativações e dicionário são estimadas minimizando a diferença entre a matriz de dados e o produto das anteriores, fixando alternadamente a matriz de ativações e o dicionário, otimizando os valores da restante matriz. As matrizes dicionário finais servem para obter estimativas de consumos parciais a partir de uma matriz de dados de consumo global. A chave para os bons resultados do algoritmo vem da regularização na otimização das matrizes de ativações. Forçar a esparsidade das matrizes de ativações facilita a obtenção de funções base mais completas; A utilização de *group lasso* [8], da diferença absoluta entre os valores observados e da média do consumo da classe de aparelhos em questão como termos de penalização na função de erro do algoritmo ajuda a evitar a ativação de classes de equipamento não presentes na casa e a encorajar a ativação em vários instantes dos equipamentos realmente presentes. Esta metodologia requer um conjunto de casas para as quais todos os aparelhos elétricos estejam monitorizados para efetuar previsões sobre casas que apenas possuam um *smartmeter*. Segundo os autores, é possível atingir bons resultados com intervalos de uma hora entre medições de consumo.

Dong, Wang e Lu publicaram, em 2013, um modelo baseado no de Kolter et al. para desagregação de consumo de água no setor doméstico [6]. Este modelo introduz um conceito de desagregação hierárquica/recursiva. A sua aplicação é comparável a um caso específico da aplicação do modelo de Kolter et al. em que, partindo do consumo global, se consideram apenas

duas classes de equipamentos para desagregação: uma classe de aparelho específica e o restante consumo da casa. Assim, é possível estimar o consumo de cada classe recursivamente partindo do consumo restante da iteração anterior. Da mesma forma que é aplicado ao consumo de água, este algoritmo é capaz de isolar sequencialmente os aparelhos elétricos que mais se destacam dos restantes em termos de consumo.

Em 2015, Batra, Singh e Whitehouse publicaram uma nova abordagem ao problema de desagregação de consumo, intitulada de *Neighbourhood NILM* [9]. *Neighbourhood NILM* é uma metodologia de agrupamento por vizinhos mais próximos que apenas utiliza infraestruturas de medição de aparelhos para um conjunto casas de treino. O objetivo do algoritmo é conseguir estimar os consumos parciais de novas casas por comparação às casas do conjunto de treino. A estimativa para o consumo de cada classe de aparelhos é a média dos valores observados do consumo dessa classe para um conjunto de "casas vizinhas"selecionadas de entre as casas do conjunto de treino. As casas vizinhas são selecionadas consoante os valores de um conjunto de covariáveis, que difere de classe para classe. As variáveis utilizadas para agrupar os clientes são variáveis tipicamente de acesso fácil como consumos globais mensais, área da casa ou número de habitantes da casa. Em 2016, os mesmo autores publicaram uma nova versão do algoritmo à qual chamaram *Gemello* [5]. A publicação inclui uma formulação matemática mais detalhada, o conjunto de variáveis independentes utilizado pelos autores (incluindo variáveis derivadas de medições de 15 em 15 minutos) e comparação da precisão com algoritmos sofisticados que se baseiam na recolha de dados de consumo com alta frequência. Embora a precisão do algoritmo apresentada seja alta, é de salientar que o estudo foi conduzido no Texas (E.U.A.). Ainda que o estado do Texas tenha climas extremamente variados, a população está concentrada na zona oriental com clima oceânico homogéneo, o que facilita a estimação do consumo, por exemplo, de aparelhos climatização através de agrupamento por vizinhos mais próximos.

2. Metodologia

2.1 Medidas de erro e precisão

Vários modelos e algoritmos são avaliados pelo seu poder preditivo neste trabalho. Para tal, são necessárias medidas de precisão ou de erro para comparar o desempenho das diferentes metodologias, pelo que se propõem as três medidas abaixo apresentadas.

Root Mean Squared Error O *RMSE* (raiz do erro quadrático médio) é a função de erro mais frequentemente utilizada em comparação de várias classes de modelos. Seja Y_i o i-ésimo valor real da variável de interesse na nova amostra e $\hat{Y}_i$ o valor predito pelo modelo em estudo para a mesma observação:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2.1}$$

Uma vez que o *RMSE* usa os quadrados dos erros, os que tiverem maior magnitude terão uma grande contribuição para a média. Esta medida é útil se a ocorrência de erros grandes for especialmente indesejável.

Mean Absolute Error O *MAE* (erro absoluto médio) é uma medida de magnitude do erro em que as contribuições dos erros são proporcionais aos seus valores absolutos.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid Y_i - \hat{Y}_i \mid \tag{2.2}$$

Apesar de o *MAE* ser mais facilmente interpretável, o *RMSE* tem a vantagem de não utilizar o valor absoluto que pode ser indesejável em alguns cálculos, por não ser uma função continuamente diferenciável [10].

***Energy Accuracy* (Batra et al.)** A medida de precisão proposta por Batra, Singh e Whitehouse em [5] está definida para ser diretamente interpretada, comparável e aplicável a problemas de desagregação de consumo de energia. No entanto, os autores efetuam um truncamento dos erros absolutos relativos pelo que a medida não reflete a existência de erros superiores a 100%. Por ser baseada em erros relativos, também tem a desvantagem de ter um comportamento errático para valores observados muito baixos (e não ser definida para valores observados

iguais a zero).

$$EnergyAccuracy = 100\% \cdot \left(1 - \frac{1}{n}\sum_{i=1}^{n} trunc\left(\frac{\mid Y_i - \hat{Y}_i \mid}{\mid Y_i \mid}\right)\right) \qquad trunc(v) = \begin{cases} v & \text{se } v \leq 1 \\ 1 & \text{se } v > 1 \end{cases} \tag{2.3}$$

2.2 Validação cruzada

A capacidade de generalização de um modelo a partir de uma dada amostra de dados pode ser fraca devido a Subajustamento (*Underfitting*) ou Sobreajustamento (*Overfitting*).

O subajustamento traduz-se, como o nome indica, no fraco ajustamento do modelo ao conjunto de dados. Este é facilmente detetável a partir de medidas de qualidade de ajustamento (como o coeficiente de determinação – R^2). É esperado que um modelo com fraco ajustamento também tenha fracas previsões para novos conjuntos de dados.

O fenómeno de sobreajustamento surge quando o modelo tem um ajustamento aparentemente bom, mas na verdade não está a capturar uma boa generalização dos dados – está a ajustar-se ao ruído dos mesmos. Os erros de previsão são bastante superiores em valor absoluto aos erros de ajustamento quando este fenómeno está presente [11].

A validação cruzada é uma metodologia de estimação dos erros de previsão (que são uma boa medida da capacidade de generalização do modelo).

Método *k-fold* O método de validação cruzada *k-fold* começa pela divisão dos dados em k sub amostras (*folds*) de igual dimensão ou de dimensão mais próxima possível. A escolha das *folds* é aleatória. O algoritmo efetua k iterações em que cada iteração i tem o seguinte processo:

- é constituído um conjunto de treino que contém todas as *folds* exceto a *fold* i;
- o conjunto de teste ou de validação é constituído pela *fold* i;
- ajusta-se o modelo ao conjunto de treino;
- obtêm-se previsões de acordo com o modelo ajustado ao conjunto de treino para o conjunto de validação;
- a partir das previsões obtidas, são calculados os erros de previsão para as observações pertencentes ao conjunto de validação.

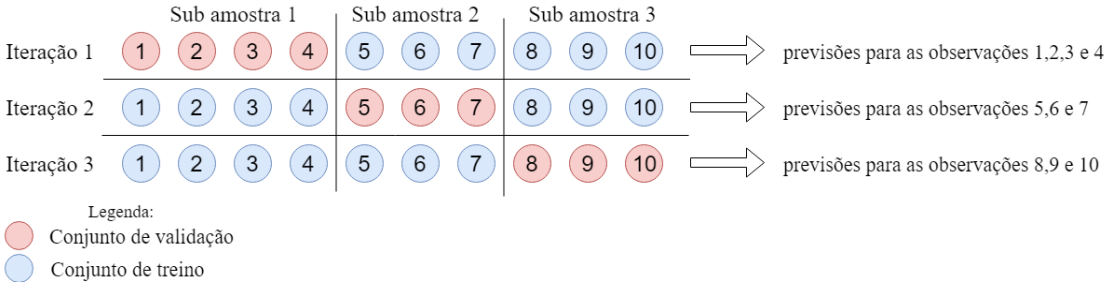


Figura 2.1: Esquema do processo de uma validação cruzada 3-fold.

Ao caso particular do método de validação cruzada *k-fold* em que k=n (número de observações na amostra) dá-se o nome de método *Leave-One-Out*.

2.3 Análise em componentes principais

A análise em componentes principais (ACP) é uma técnica de análise de dados multivariados que procura transformar um conjunto de dados com um elevado número de variáveis aleatórias num número reduzido de combinações lineares das variáveis aleatórias originais. O objetivo é reduzir a dimensão dos dados, preservando a maior porção possível da variabilidade total da amostra original [12]. A estas combinações lineares das variáveis originais atribui-se o nome de componentes principais.

Seja X_i a i-ésima de p variáveis, Y_j a j-ésima componente principal e $\mathbf{X}$=$[X_1\ X_2\ ...\ X_p]^T$.

$$Y_j = a_j^T \cdot \mathbf{X} \tag{2.4}$$

O primeiro passo é determinar o vetor a_1 para obter a primeira componente principal (Y_1). a_1 obtém-se maximizando a variância de $a_1^T \cdot \mathbf{X}$. No entanto, sem acrescentar restrições, os valores de a_1 poderiam não ser finitos. Então acrescenta-se a condição:

$$a_j^T \cdot a_j = 1, \quad \forall j \in \{1, ..., p\} \tag{2.5}$$

O vetor a_2 é calculado maximizando a variância de $a_2^T \cdot \mathrm{X}$, sujeito à restrição (2.5) e adicionalmente sujeito a que a covariância entre $a_2^T \cdot \mathbf{X}$ e $a_1^T \cdot \mathbf{X}$ seja nula. Analogamente o cálculo dos vetores a_k^T (para k>1) é sujeito a:

$$cov(a_k^T \cdot \mathbf{X}, a_j^T \cdot \mathbf{X}) = 0, \quad \forall j \in \{1, ..., k-1\} \tag{2.6}$$

Uma propriedade importante das componentes principais é que cada componente principal tem variância superior às componentes que foram obtidas posteriormente:

$$Var(Y_1) > Var(Y_2) > ... > Var(Y_p) \tag{2.7}$$

Como o objetivo do procedimento é reter poucas componentes principais mantendo o máximo de informação possível, normalmente retêm-se as m primeiras componentes principais, que são as m componentes com maior variância (de preferência com $m<<p$).

Na maioria das aplicações nem todas as variáveis terão a mesma escala ou a mesma natureza, ou pelo menos algumas serão de natureza desconhecida. Para evitar que sobressaiam as variáveis que têm uma escala maior e se perca informação sobre as restantes, é frequentemente recomendado que se padronize as variáveis originais antes de iniciar o processo de cálculo das componentes principais.

Realisticamente, salvo em raras exceções, o conjunto de dados $\mathbf{X}$ não contém a totalidade da população em estudo, mas sim uma amostra proveniente desta. Sendo assim as as variáveis obtidas através deste processo são, na verdade, as estimativas das componentes principais ($\hat{Y}_j$). Estas são frequentemente chamadas de componentes principais por uma questão de simplicidade.

2.4 Modelos de regressão linear

Um modelo de regressão é um modelo que pretende explicar a variabilidade de uma ou mais variáveis de interesse através de variáveis explicativas.

As variáveis de interesse (ou de resposta) são representadas pela letra "Y"e as variáveis explicativas são representadas pela letra "X". Considera-se que a variável de resposta pode ser

decomposta na soma de duas componentes: a componente sistemática e a componente aleatória (2.8).

$$Y = \underbrace{r(X_1, ..., X_p)}_{componente\ sistemática} + \underbrace{\varepsilon}_{componente\ aleatória} \tag{2.8}$$

Regressão Linear Se $r(.)$ for uma função linear nos seus parâmetros (representados pela letra "β"), o modelo é de regressão linear e a equação (2.8) assume a forma de (2.9).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon \tag{2.9}$$

A regressão linear é um conceito que existe desde antes da era dos computadores mas, ainda assim, existem razões para continuar a ser usada. Além de ser um modelo que permite uma interpretação fácil da relação entre variáveis de interesse e explicativas, é capaz de produzir previsões melhores que modelos não lineares bastante mais complexos quando o número de observações disponíveis é reduzido [8].

No modelo de regressão linear a variável resposta para cada indivíduo i tem a seguinte expressão (2.10):

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \varepsilon_i \tag{2.10}$$

Supõe-se que as v.a.s ϵ_i são independentes e identicamente distribuídas, com distribuição $N(0, \sigma)$ [13].

Uma vez que o valor médio do erro aleatório é nulo, o valor esperado da variável resposta é a componente sistemática do modelo, cujo estimador é (2.11):

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot X_1 + ... + \hat{\beta}_p \cdot X_p \tag{2.11}$$

As estimativas dos parâmetros β_j, representadas por $\hat{\beta}_j$, são obtidas minimizando uma função dos resíduos, e_i (2.12).

$$e_i = y_i - \hat{y}_i \tag{2.12}$$

Mínimos Quadrados O método de estimação dos parâmetros mais amplamente utilizado é o Método dos Mínimos Quadrados que consiste na minimização da soma dos quadrados dos resíduos (*RSS – Residual Sum of Squares*). O desenvolvimento do Método dos Mínimos Quadrados é atribuído a Carl Friedrich Gauss em 1795, embora a primeira publicação sobre a metodologia tenha sido da autoria de Adrien-Marie Legendre em 1805 [14].

$$RSS(\beta_0, \beta_1, ..., \beta_p) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{2.13}$$

A Figura 2.2 ilustra a aplicação do Método dos Mínimos Quadrados considerando uma variável explicativa e uma variável resposta.

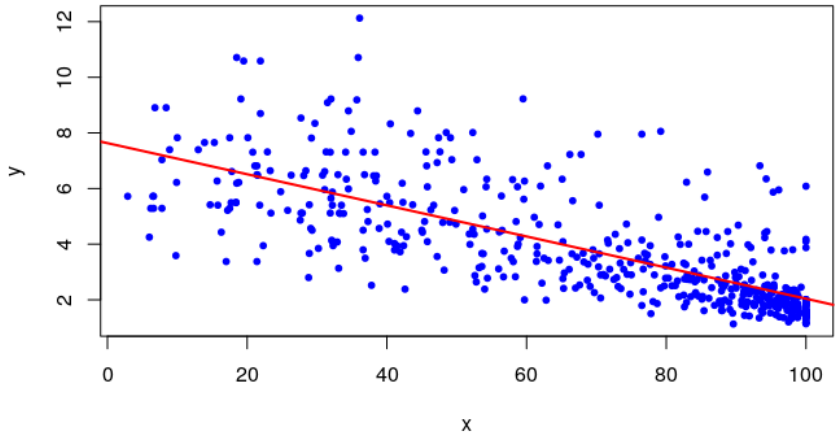


Figura 2.2: Representação gráfica do ajustamento de um modelo cujos parâmetros foram estimados pelo Método dos Mínimos Quadrados.

Mínimos Quadrados Aparados O Método dos Mínimos Quadrados Aparados (*Least Trimmed Squares*) é uma alternativa robusta ao método dos mínimos quadrados. Consiste na estimação dos parâmetros a partir da minimização da soma aparada dos quadrados dos resíduos (*Trimmed RSS*), que exclui os resíduos de maior magnitude. Seja $e^2_{(i)}$ a estatística ordinal de ordem i dos quadrados dos resíduos.

$$TRSS = \sum_{i=1}^{h} e^2_{(i)} \tag{2.14}$$

O hiperparâmetro h deverá estar localizado entre $\frac{n}{2}$ e n (não inclusive), sendo o valor mais frequentemente utilizado $h = \left([\frac{n}{2}] + [\frac{p+1}{2}]r\right)$ [15].

Transformação Box-Cox Em alguns casos o pressuposto de normalidade dos erros não se verifica aquando da análise de resíduos. Nestas circunstâncias, o curso comum é efetuar uma transformação sobre a variável resposta e ajustar um novo modelo. A transformação Box-Cox é uma transformação sobre a variável resposta que visa obter uma nova resposta com distribuição Normal [16].

$$Y' = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{se } \lambda \neq 0 \\ log(Y) & \text{se } \lambda = 0 \end{cases} \tag{2.15}$$

O parâmetro λ da transformação é estimado através do método da máxima verosimilhança [16].

Modelos Lineares Generalizados Através de uma generalização do modelo linear, permite-se que a distribuição dos erros assuma um modelo não Normal. Seja Y a variável aleatória de interesse original, utiliza-se uma função de ligação $g(.)$ para se obter um preditor linear (2.16).

$$g(E(Y)) = \eta = \beta_0 + \beta_1 \cdot X_1 + ... + \beta_p \cdot X_p \tag{2.16}$$

As previsões para a variável de interesse são obtidas aplicando o inverso da função ligação aos valores preditos pelo preditor linear. No presente trabalho foi usado o logaritmo como função de ligação para erros com distribuição de probabilidade Gama.

2.5 Agrupamento pelos vizinhos mais próximos

O método de agrupamento pelos k vizinhos mais próximos (em inglês: *k-Nearest Neighbors* ou *kNN*) é um método não paramétrico que produz estimativas para uma variável ou vetor aleatório de interesse, não através de uma generalização em forma de um modelo, mas sim utilizando informação armazenada sobre indivíduos da população em estudo. É um método não paramétrico pois não requer pressupostos sobre a distribuição de probabilidade da população subjacente aos dados. A primeira aplicação deste método é frequentemente atribuída a Fix e Hodges [17], embora o conceito tenha evoluído desde o século XI com Alhazen [18].

O método consiste em encontrar k indivíduos, para os quais temos informação sobre uma variável de interesse, que estejam "próximos"de um indivíduo para o qual se pretende estimar o valor da mesma variável. No contexto de um problema de classificação, define-se frequentemente que cada um destes k vizinhos tenha um "voto"na classificação do indivíduo em estudo. Já no caso do problema de regressão com resposta quantitativa, a estimativa do valor da variável de interesse para o indivíduo seria, habitualmente, calculada através de uma média aritmética ou ponderada sobre os valores referentes aos k vizinhos.

O conceito de vizinho mais próximo exige a existência de uma medida de distância ou proximidade que seja uma função de variáveis independentes. Uma função de distância entre dois elementos/indivíduos deve tomar valores exclusivamente em $\mathbb{R}_0^+$ com mínimo em 0 (quando os vetores de covariáveis são idênticos para os dois elementos). É possível obter uma função de distância através de uma função de proximidade e vice-versa através da relação:

$$dist(I, I') = \frac{1}{prox(I, I')} \tag{2.17}$$

Se uma função de proximidade *prox* e uma função de distância *dist* satisfazem a equação (2.17) então o procedimento que determina o conjunto de vizinhos mais próximos de I através da minimização de $dist(I, I')$ é equivalente ao procedimento que se baseia na maximização de $prox(I, I')$.

A função de distância mais comum é a distância euclidiana. Abaixo está descrita uma formulação do método de agrupamento pelos k vizinhos mais próximos para o problema com resposta quantitativa:

- Seja $\mathbf{X}$=$[X_1\ X_2\ ...\ X_p]^T$ um vetor aleatório (de covariáveis) e Y a variável aleatória de interesse para o problema.
- Seja G um conjunto de observações cujos valores observados de Y e do vetor de covariáveis $\mathbf{X}$ são conhecidos e estão guardados.
- Seja $dist(I, I') \equiv dist(\mathbf{x}, \mathbf{x'})$ a função de distância entre duas observações/indivíduos I e I' com concretizações do vetor de covariáveis $\mathbf{x}$ e $\mathbf{x'}$, respetivamente.

Para um dado indivíduo I, o seu conjunto de k vizinhos mais próximos, V, é sujeito a:

$$V \subset G \tag{2.18}$$

$$\forall I' \in V, \forall I'' \in G \backslash \{V\} : \quad dist(I, I') \leq dist(I, I'') \tag{2.19}$$

A estimativa do valor da variável de interesse, $\hat{y}$, é obtida através do seguinte estimador:

$$\hat{Y} = \sum_{i \in V} w_i \cdot Y_i \tag{2.20}$$

w_i é o peso dado à observação i na estimativa do valor de Y para o indivíduo I. No caso mais comum, $w_i = \frac{1}{k}$, o estimador é a média aritmética dos valores observados de Y para os k vizinhos mais próximos de I.

Os pesos w_i, no caso da média ponderada, são calculados com base numa função decrescente da distância. Uma mais valia desta variante é que, uma vez que as observações têm uma ponderação associada, não há desvantagem em considerar todas as observações em estudo em vez de considerar apenas k, eliminando a necessidade de determinar qual o melhor valor para este parâmetro. No entanto esta abordagem levanta o problema da escolha da função do peso e pode aumentar bastante o custo computacional do algoritmo para grandes amostras.

2.6 Redes neuronais

Redes neuronais artificiais, também conhecidas como modelos de *Deep Learning*, são modelos cuja popularidade tem sido notável nos últimos anos, devido à sua frequente aplicação na área da inteligência artificial. Este tipo de modelo caracteriza-se por ser versátil na medida em que é aplicável a vários tipos de problemas de modelação como classificação (com duas ou mais classes) e regressão com variáveis dependentes reais (seja simples ou múltipla, com resposta univariada ou multivariada). Apesar da euforia da comunidade de *Data Science* relativamente a esta classe de modelos, as redes neuronais são apenas modelos estatísticos não lineares [8].

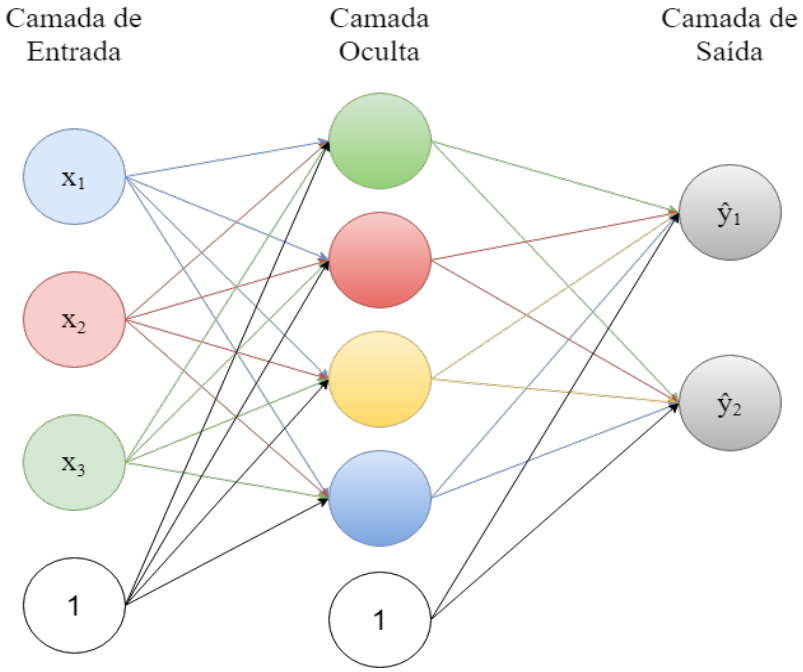


Figura 2.3: Diagrama representativo de uma rede neuronal com apenas uma camada oculta.

A estrutura de uma rede neuronal é composta por três tipos de vértice:

- Vértices de entrada (*input layer*), que recebem valores observados de variáveis independentes e são o ponto de partida da rede.

- Vértices ocultos (*hidden layers*), onde se efetuam cálculos intermédios do processo preditivo;
- Vértices de saída (*output layer*), onde se obtêm as previsões para as variáveis dependentes.

A obtenção de previsões para as variáveis de interesse com base em valores observados das variáveis independentes traduz-se no seguinte processo:

- Os vértices de entrada são inicializados com valores observados das variáveis independentes;
- Esta informação passa por uma ou mais camadas de vértices ocultos, sofrendo transformações em cada arco e vértice que atravesse;
- Os resultados destas transformações sucessivas são os valores que se obtêm nos vértices de saída no final do processo. Cada vértice de saída terá associado o valor da previsão para uma das variáveis de resposta.

Pesos e funções de ativação Cada arco da rede tem um parâmetro de peso associado pelo qual se multiplica o valor que o arco transmite ao vértice seguinte (geralmente representado pela letra "w"). O valor recebido pelo vértice seguinte será uma soma dos valores transmitidos pelos arcos incidentes no vértice multiplicados pelos pesos correspondentes. Na representação da Figura 2.3, foram acrescentados vértices com o valor 1 para adicionar um termo independente a estas somas. O valor final de um vértice é obtido após a aplicação de uma função de ativação $f(v)$ ao valor recebido (Figura 2.4).

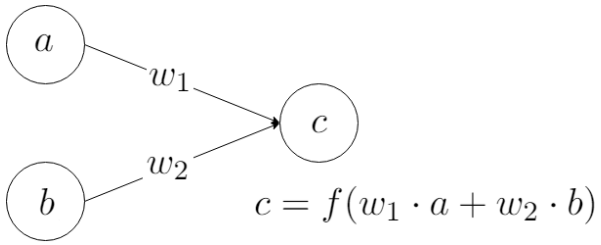


Figura 2.4: Representação do modo de transmissão de informação de uma rede neuronal.

A função de ativação $f(v)$ é responsável pela introdução de uma componente não linear no modelo. Uma vez que uma combinação linear de combinações lineares continua a ser uma combinação linear, se a função de ativação for linear para todos os vértices (por exemplo $f(v) = v$) a rede neuronal será um modelo linear.

As funções de ativação são escolhidas *a priori*, com base no tipo de resposta esperado. A função de ativação mais comum é a função logística padrão (2.21). A utilização conjunta da função logística como função de ativação e dados normalizados para o intervalo [0,1] resulta, frequentemente, em melhores modelos que a utilização dos dados inalterados.

$$f(v) = \frac{1}{1 + e^{-v}} \tag{2.21}$$

Uma vez que a função de ativação é escolhida *a priori*, o ajustamento do modelo aos dados vai depender das estimativas para os pesos $\mathbf{w}$ associados aos arcos.

Para estimar os parâmetros do modelo, os pesos $\mathbf{w}$, é necessária a existência de uma função de perda (ou função de erro) para indicar a qualidade de ajustamento do modelo aos dados. A função mais frequentemente usada no caso da resposta com suporte em $\mathbb{R}_0^+$ é o *RMSE*. As estimativas dos pesos $\mathbf{w}$ são obtidas através de um algoritmo chamado *backpropagation*.

Backpropagation: é o algoritmo iterativo que atualiza os pesos **w** em cada iteração para reduzir o *RMSE* total passo a passo. É chamado de propagação para trás por (em cada iteração) atualizar os pesos camada a camada, começando pela última camada de arcos (incidentes nos vértices de saída) e acabando nos arcos que têm origem nos vértices de entrada. O peso de cada arco é atualizado conforme o valor da derivada parcial do erro em ordem ao peso em questão e um parâmetro η de taxa de aprendizagem (2.22).

$$w_i^* \leftarrow w_i - \eta \cdot \frac{\partial RMSE_{total}}{\partial w_i} \tag{2.22}$$

O algoritmo pode ser parado quando a função de erro convergir para um mínimo. No entanto, não é desejável que o algoritmo atinja o mínimo global do *RMSE* porque, no caso geral, seria sinal de sobreajustamento [8].

2.7 Árvores de decisão

Árvore de decisão é uma classe de modelos preditivos que se caracteriza por fazer previsões para a variável resposta através da tomada de decisões sucessivas. Estes modelos são chamados árvores de decisão por poderem ser representados por uma rede em forma de árvore (em que cada decisão é uma ramificação).

As ramificações são feitas através dos valores das variáveis independentes (Figura 2.5).

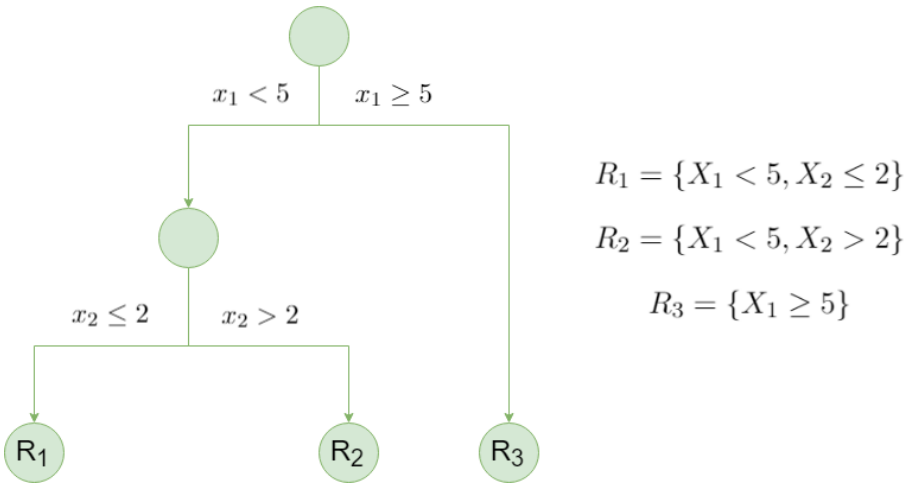


Figura 2.5: Exemplo de representação de uma árvore de decisão com duas variáveis independentes (X_1 e X_2).

Se a variável resposta for quantitativa, a estimativa para um indivíduo que se encontre numa região R_m é a média dos valores da variável resposta das observações nessa região (2.23) .

$$\hat{Y}_{\{X_1,\ldots,X_p\}\in R_m} = \overline{Y}_{\{X_1,\ldots,X_p\}\in R_m} \tag{2.23}$$

No caso em que a variável resposta é qualitativa, a estimativa da probabilidade de que um indivíduo pertença a uma classe k se está localizado na região R_m corresponde à proporção de observações da região R_m que pertence à classe k.

Encontrar a melhor partição binária (em termos de minimização da soma dos quadrados dos erros) é, geralmente, um problema computacionalmente impossível. Assim, é utilizado um algoritmo "guloso"(*greedy*) para encontrar a variável e o ponto de ramificação cujas estimativas para as duas regiões resultantes minimizem a soma dos quadrados dos erros [8]. Encontrada a

variável para a primeira decisão, são escolhidas (de forma análoga) de entre as restantes quais serão as utilizadas nas ramificações seguintes e assim sucessivamente.

2.8 *Ensemble Learning*

Ensemble Learning é uma metodologia que consiste em ajustar um modelo de previsão que combina um conjunto de modelos base resultando numa única previsão. O objetivo desta combinação é capturar os diversos pontos fortes de cada modelo em apenas um.

2.8.1 Florestas Aleatórias

Uma floresta aleatória (do inglês *random forest*) é um método que consiste em ajustar árvores de decisão a amostras *Bootstrap* obtidas a partir da amostra original e agregar as previsões destas numa previsão conjunta. *Bootstrap* é um método de reamostragem que consiste em selecionar aleatoriamente e com reposição elementos da amostra original para formar uma nova amostra.

A floresta aleatória é uma classe de metodologias de *Boostrap Aggregating*, ou *Bagging*. Esta é uma metodologia que pode reduzir drasticamente a variância de procedimentos instáveis como as árvores de decisão [8].

2.8.2 *Gradient Boosting*

Gradient Boosting é uma metodologia de *Ensemble Learning* que pertence à classe de algoritmos designada de *Boosting*. O conceito de *Boosting* é o de iterar sobre um modelo fazendo-o evoluir em cada iteração de forma a reduzir uma função de perda. Estas iterações são feitas com base nos gradientes da função de perda em ordem às previsões do modelo. No caso particular em que a função de perda seja a soma dos quadrados dos resíduos (RSS) ou a média dos resíduos quadrados, estes gradientes correspondem aos resíduos do modelo [8]. Abaixo está descrito o processo iterativo de *Gradient Boosting* para este caso particular [19]:

- Iteração 0: Seja Y a variável resposta e $\mathbf{X}$ o vetor de covariáveis; As previsões para a variável Y nesta iteração são produzidas pela função de previsão $F_0(\mathbf{X}) = \hat{F}_0(\mathbf{X}) = 0$;
- Iteração 1: Ajusta-se um modelo, tomando os resíduos da iteração anterior ($e_0 = y - \hat{F}_0(\mathbf{x})$) como observações da variável resposta. Seja $h_1(\mathbf{X})$ a função de previsão associada ao modelo tal que: $e_0 = h_1(\mathbf{X}) + \varepsilon_1$;
- As previsões para a variável resposta Y na iteração 1 são dadas por $\hat{F}_1(\mathbf{X}) = \hat{F}_0(\mathbf{X}) + \hat{h}_1(\mathbf{X})$;
- Iteração 2: Ajusta-se um novo modelo, tomando os resíduos da iteração anterior ($e_1 = y - \hat{F}_1(\mathbf{x})$) como observações da variável resposta. Seja $h_2(\mathbf{X})$ a função de previsão associada ao novo modelo tal que: $e_1 = h_2(\mathbf{X}) + \varepsilon_2$;
- As previsões para a variável resposta Y na iteração 2 são dadas por $\hat{F}_2(\mathbf{X}) = \hat{F}_1(\mathbf{X}) + \hat{h}_2(\mathbf{X})$;
- (...)
- Iteração m: Ajusta-se um novo modelo, tomando os resíduos da iteração anterior ($e_{m-1} = y - \hat{F}_{m-1}(\mathbf{x})$) como observações da variável resposta. Seja $h_m(\mathbf{X})$ a função de previsão associada ao novo modelo tal que: $e_{m-1} = h_m(\mathbf{X}) + \varepsilon_m$;

- As previsões para a variável resposta Y na iteração m são dadas por $\hat{F}_m(\mathbf{X}) = \hat{F}_{m-1}(\mathbf{X}) + \hat{h}_m(\mathbf{X})$.

O hiperparâmetro m desta metodologia é, geralmente, escolhido separando o conjunto de observações de treino em dois e utilizando um dos subconjuntos como conjunto de validação para vários valores de m (escolhendo-se o m que minimiza uma função dos erros) [8].

shrinkage Na prática, é comum a utilização um coeficiente γ de *shrinkage* (encolhimento ou taxa de aprendizagem) em cada iteração para desacelerar a convergência do algoritmo. A função de previsão para a iteração k ($k = \{1, ..., m\}$) passa a ser definida por (2.24):

$$F_k(\mathbf{X}) = F_{k-1}(\mathbf{X}) + \gamma_k \cdot h_k(\mathbf{X}), \qquad 0 < \gamma_k < 1 \tag{2.24}$$

Esta prática ajuda a evitar o sobreajustamento e permite a formação de um conjunto de modelos de maior dimensão. As previsões destes modelos são combinadas na previsão final $F_m(\mathbf{X})$.

2.8.3 *Ensemble Stacking*

Stacking é uma metodologia de *Ensemble Learning* que por vezes é designada de *Meta Ensemble* ou de modelo de segundo nível. A particularidade de um modelo de segundo nível é que o conjunto de variáveis independentes inclui valores preditos pelos modelos de primeiro nível (Figura 2.6).

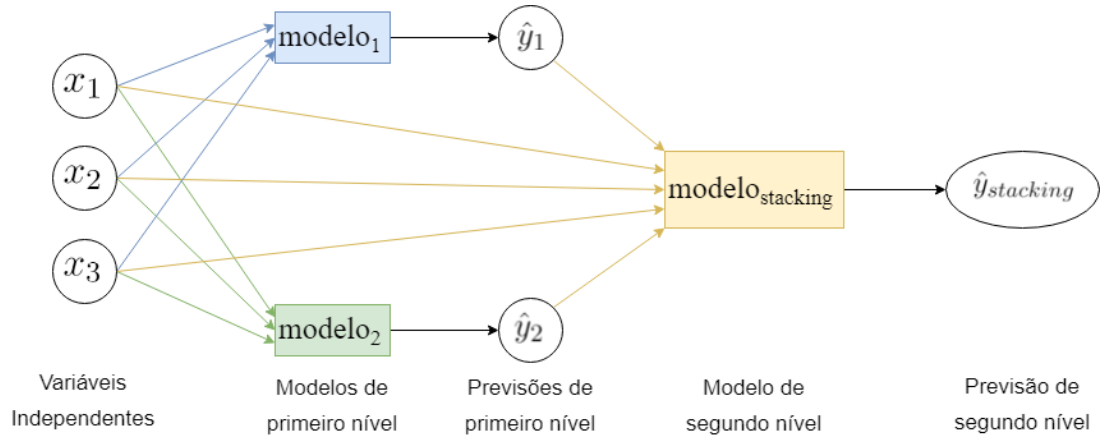


Figura 2.6: Representação do processo de estimação por *Stacking*, com dois modelos de primeiro nível($modelo_1$ e $modelo_2$) e três covariáveis (x_1,x_2 e x_3).

As previsões dos modelos de primeiro nível são obtidas a partir de uma validação cruzada *k-fold*. Desta forma é possível utilizar o mesmo conjunto de observações para ajustar os modelos de primeiro e segundo nível. A escolha da classe de modelo para o segundo nível é livre, independentemente dos modelos utilizados no primeiro nível.

Esta metodologia gera modelos de difícil interpretação e tem um custo computacional potencialmente elevado. No entanto, é uma forma eficaz de obter um modelo com maior poder preditivo. Não seria inédita a utilização de modelos com nível de ordem superior a 2, no entanto o custo computacional cresce com cada nível acrescentado.

No *Kaggle* – plataforma *Online* de concursos de *Data Science* [20] – as competições têm como objetivo construir o modelo com o maior poder preditivo e contam com a participação de

milhares de equipas. A grande maioria destas competições é ganha por equipas que recorrem a *Ensemble Stacking*, por vezes com modelos de terceiro ou quarto nível.

3. Descrição e análise dos dados

3.1 Os dados

Os dados analisados no decorrer do estágio são, como já foi mencionado anteriormente, dados relativos aos clientes do projeto EDP *re:dy*. Todos os dados de clientes utilizados estão completamente anonimizados.

3.1.1 A Base de Dados

As interações com a base de dados são processadas num ecossistema *Apache Hadoop* [21], que permite a extração de dados através da linguagem de pesquisa declarativa *SQL - Structured Query Language*. Apresenta-se, na Figura 3.1, um esquema parcial das tabelas e atributos da base de dados (incluindo apenas as tabelas e atributos considerados relevantes para a análise dos dados).

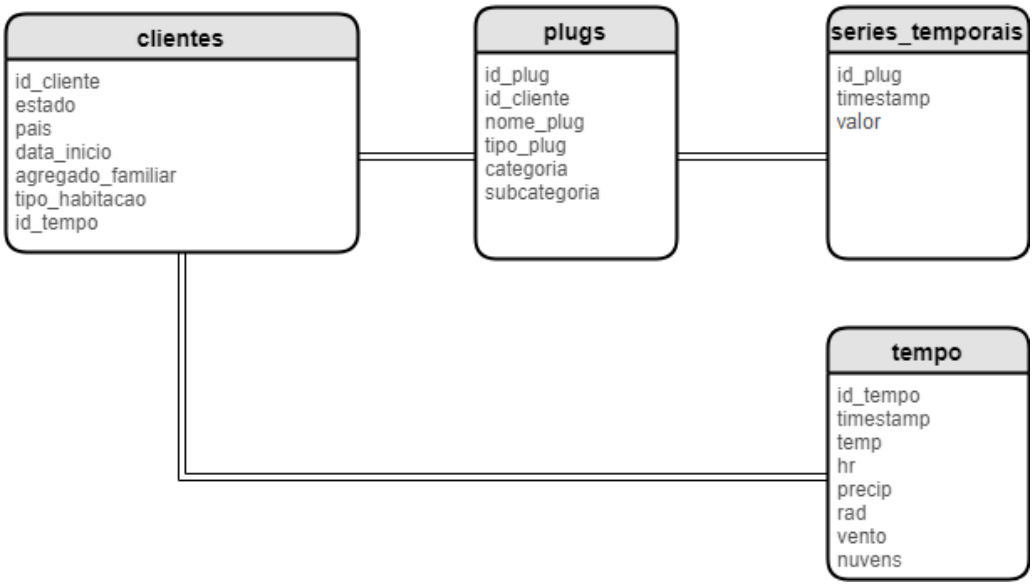


Figura 3.1: Esquema parcial da base de dados.

A tabela "clientes", como o nome indica, inclui informação sobre os clientes e as suas habitações. A tabela "plugs"contém informações sobre *smartmeters*, *plugs* e aparelhos elétricos associados. Os dados de consumo de cada *plug*/aparelho ou *smartmeter* ao longo do tempo estão contidos na tabela "series_temporais". Na tabela "tempo"está armazenada informação sobre as condições meteorológicas de diversas zonas geográficas ao longo do tempo. Os atributos apresentados na Figura 3.1 estão descritos na Tabela 3.1.

Tabela 3.1: Descrição dos atributos da base de dados.

Atributos de ligação entre as tabelas

atributo	tabelas	descrição
id_cliente	clientes, plugs	Identificador anonimizado de cliente.
id_plug	plugs, series_temporais	Identificador de *plugs* ou *smartmeters*.
id_tempo	clientes, tempo	código numérico da zona do cliente.
categoria	plugs	categoria do aparelho (código numérico).
subcategoria	plugs	subcategoria do aparelho (código numérico) .
pais	clientes	país de residência do cliente.
data_inicio	clientes	data de início dos serviços do *re:dy* para o cliente.
agregado_familiar	clientes	número de pessoas no agregado familiar do cliente.
tipo_habitacao	clientes	tipo da habitação do cliente (Vivenda ou Apartamento).
nome_plug	plugs	nome do aparelho elétrico (digitado pelo cliente).
tipo_plug	plugs	tipo de aparelho (distingue *smart-meters* das *plugs*).
timestamp	series_temporais	instante de leitura do valor da série temporal.
timestamp	tempo	instante da previsão meteorológica.
temp, hr, precip, rad, vento, nuvens	tempo	previsões meteorológicas relativas a: temperatura, humidade relativa, precipitação, radiação, velocidade do vento e nebulosidade (respetivamente).
valor	series_temporais	consumo de elétricidade (em kWh) do aparelho lido no dado instante (timestamp).

Cada aparelho/*plug*, representado por uma linha na tabela "plugs", tem um código numérico de categoria (entre 0 e 9) e subcategoria (entre 0 e 4). A correspondência destes códigos com respetivas tipologias de aparelho está descrita na Tabela 3.2.

Tabela 3.2: Categorias e subcategorias disponíveis para classificação das *plugs*.

	Subcategorias				
Categorias	0	1	2	3	4
0.Outros	Outros	-	-	-	-
1.Cozinha	Outros	Forno	Micro-ondas	-	-
2.Refrigeração	Outros	Frigorífico	Combinado	Arca	-
3.Máquinas	Outros	Máquina da Loiça	Máquina de Lavar	Máquina de Secar	-
4.Multimédia	Outros	Televisor	Leitor DVD	Box	Consola de Jogos
5.Informática	Outros	Computador	Monitor	Impressora	Portátil
6.Iluminação	Outros	Incandescente	Halogénio	Fluorescente	LED
7.Aquecimento	Outros	Acumulador de Calor	Aquecedor a Óleo	Termoventilador	-
8.Arrefecimento	Outros	Ar Condicionado	-	-	-
9.Águas quentes sanitárias	Outros	Termoacumulador	-	-	-

3.1.2 Limitações

Abaixo enumeram-se as principais limitações dos dados do projeto *re:dy*:

1. Apenas cerca de um terço dos clientes *re:dy* tem informação acerca do tamanho do seu agregado familiar e do tipo de habitação.

2. As séries temporais de consumo associadas a *plugs* apresentam, por vezes, falhas devido a problemas de comunicação sem fios.

Grande parte dos clientes *re:dy* possui painéis solares e mede a sua produção de energia através de *plugs*. Tal como as *plugs* de consumo, por vezes estas apresentam erros de comunicação, o que é um problema, uma vez que o valor da energia produzida pelo cliente é necessária para o cálculo do seu consumo global.

3. Os clientes têm a opção de trocar as *plugs* de aparelho em aparelho. Embora se possa trocar a categoria da *plug* para corresponder a um novo aparelho elétrico, nem a data de alteração nem a categoria anterior ficam guardadas na base de dados. Nestas circunstâncias é possível encontrar, por exemplo, um equipamento categorizado como um frigorífico mas que tem consumo correspondente a uma televisão durante meses de histórico (sem qualquer informação de que houve uma alteração).

4. As *plugs* são categorizadas pelo próprio cliente. Uma vez que as *plugs* têm um título além da categoria atribuída pelo cliente, o cliente pode não sentir necessidade de lhes atribuir categorias ou de as alterar devidamente quando troca a *plug* para outro aparelho. Existem, por isto, clientes sem categorização de equipamentos ou com equipamentos mal categorizados.

5. No ínicio do estágio não existia uma lista de aparelhos elétricos do cliente, pelo que não havia informação sobre a quantidade de aparelhos de cada categoria que este tenha. Exemplo: se o cliente tiver *plugs* em dois aquecimentos, nada garante que estes sejam os seus únicos aparelhos de climatização.

   A falta de informação sobre a lista de aparelhos elétricos do cliente é um dos maiores obstáculos à desagregação de consumos por tipologia, pois faz com que o número de aparelhos de uma certa categoria que o cliente mede com *plugs* seja potencialmente inferior ao número de aparelhos que o cliente realmente possui/utiliza.

   Uma funcionalidade recente da aplicação para *smartphones* do EDP *re:dy* (adicionada em junho de 2018) permite ao cliente listar que tipo de equipamentos possui, embora não garanta que o cliente finalize a sua caracterização completamente. Na altura em que o trabalho foi realizado, a fração de clientes que tinha caracterizado os seus equipamentos ainda era pequena.

Para ultrapassar a limitação 5, a modelação no presente estudo concentra-se maioritariamente nas categorias/subcategorias de aparelhos elétricos cuja ocorrência é de uma unidade por agregado familiar (salvo raras exceções): frigoríficos/combinados, máquinas de lavar roupa e máquinas de lavar loiça.

3.1.3 A Amostra

A primeira amostra de clientes utilizada no estágio continha 209 clientes selecionados aleatoriamente de entre os clientes *re:dy* (cerca de 19000). Após uma pequena análise aos dados destes 209 clientes concluiu-se que a amostra não seria suficiente devido ao reduzido número de *plugs* com categoria de aparelho elétrico atribuída.

Com o objetivo de encontrar observações mais informativas, foi selecionada uma amostra diferente. A segunda amostra foi selecionada de entre os clientes que cumprem os seguinte requisitos:

- é cliente ativo;

- é cliente residente em Portugal;
- tem data de adesão anterior ao ano de 2017.

Para cada cliente, calculou-se a proporção do consumo global do último mês de 2017 que estava medida por *plugs* classificadas. Foram selecionados os 400 clientes para os quais esta proporção era maior.

Apenas cerca de um terço dos clientes da nova amostra tinham informação sobre o tamanho do seu agregado familiar. Uma vez que em abordagens anteriores [5, 9] ao problema da desagregação de consumos se verificou que esta variável teria utilidade na previsão dos consumos parciais, acrescentaram-se mais clientes à amostra. Dos que não pertenciam aos 400 selecionados anteriormente, foram escolhidos todos aqueles que cumprissem os seguinte requisitos:

- é cliente ativo, residente em Portugal e com data de adesão anterior ao ano de 2017;
- o número de pessoas no agregado familiar do cliente é conhecido;
- tem pelo menos uma *plug* identificada com um dos seguintes tipos de aparelho: frigorífico, combinado, máquina de lavar roupa ou máquina de lavar loiça.

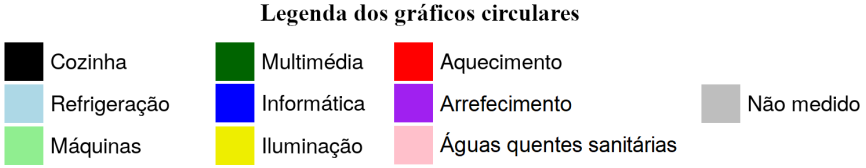
Foram acrescentados clientes com *plugs* em frigoríficos e/ou máquinas, de forma a aumentar a amostra disponível para a modelação destes consumo parciais. Selecionaram-se apenas clientes com informação sobre o tamanho do agregado familiar, uma vez que estudos anteriores [5, 9] consideraram esta variável importante para a desagregação do consumo. Acrescentando estes clientes aos 400 já selecionados obteve-se uma amostra final para análise com 526 clientes.

3.2 Análise dos dados

3.2.1 Gráficos circulares

A primeira representação gráfica dos dados dos clientes obtida foram gráficos circulares relativos aos consumos mensais de cada cliente. Estes gráficos apresentam a informação de desagregação de consumo que os clientes já possuem, ou seja, a fração de consumo das suas *plugs* face ao consumo global. Esta representação está diretamente ligada ao objetivo do estágio, pois representa a informação de consumos parciais atualmente apresentada ao cliente, a qual se pretende complementar através da desagregação de consumos. Por uma questão de simplicidade, ao invés de apresentar os 12 gráficos mensais para cada cliente, apresenta-se um gráfico por trimestre e respetivos consumos globais totais.

Os clientes foram numerados de 1 a 526 anteriormente à análise para facilitar a sua identificação.



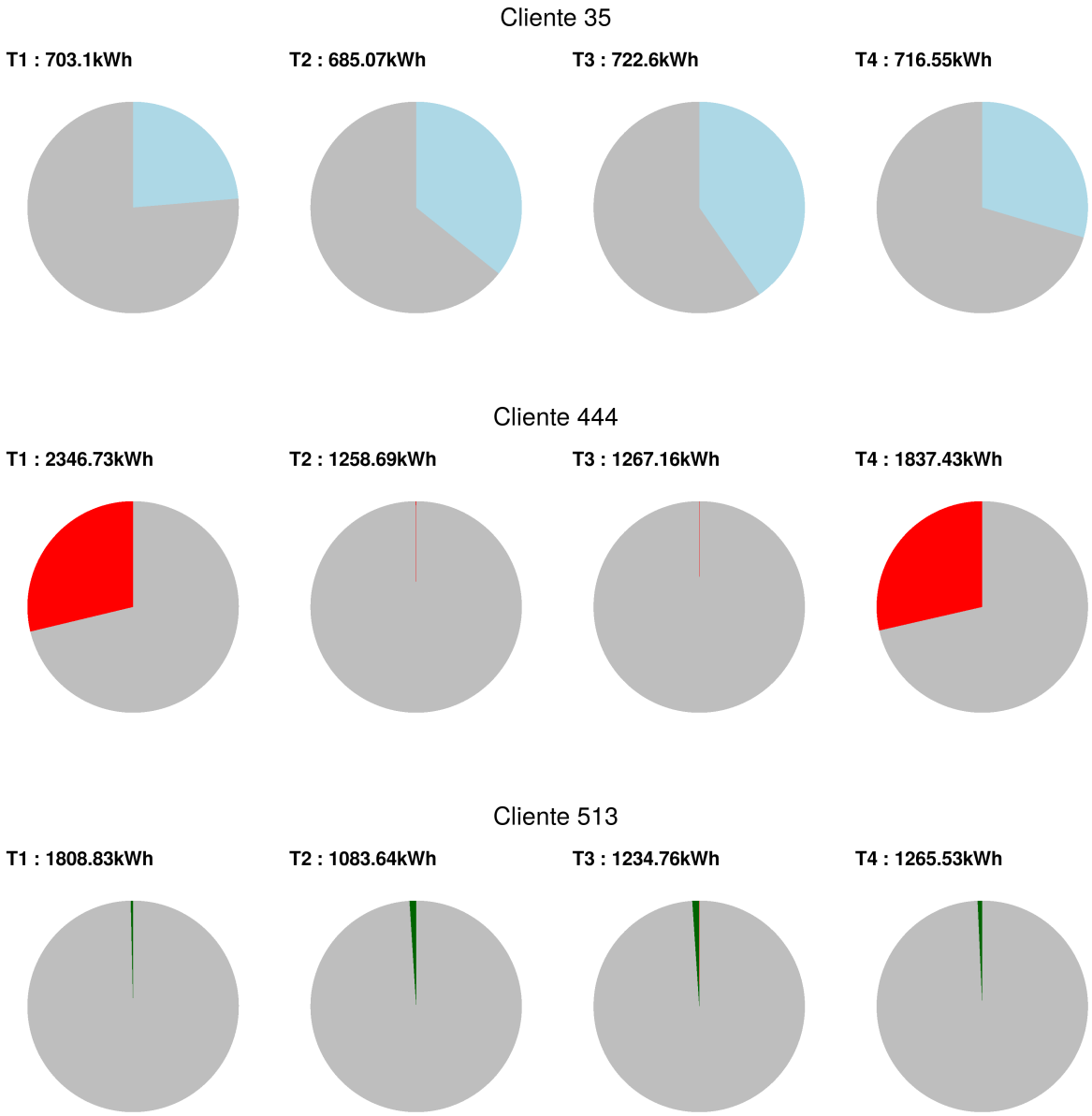


Figura 3.2: Gráficos circulares representativos dos consumos das *plugs* dos clientes 35, 444 e 513 face aos seus consumos globais, para o ano de 2017.

Os gráficos apresentados na Figura 3.2 são alguns dos que se consideraram ilustrativos da amostra em estudo. A grande maioria dos clientes da amostra é composta por clientes donos de apenas uma *plug* como os clientes 35, 444 e 513. O gráfico do cliente 35 realça a sazonalidade anual dos frigoríficos, cujo consumo é maior nas alturas do ano em que as temperaturas são mais elevadas. O cliente 444 demonstra a sazonalidade acentuada do aquecimento ambiente. Apesar destes dois clientes demonstrarem a sazonalidade habitual destas categorias de equipamentos, esta não se verifica em todos os clientes. O cliente 513 é um exemplo de cliente com consumo global alto e *plugs* que pouco contribuem para a sua compreensão.

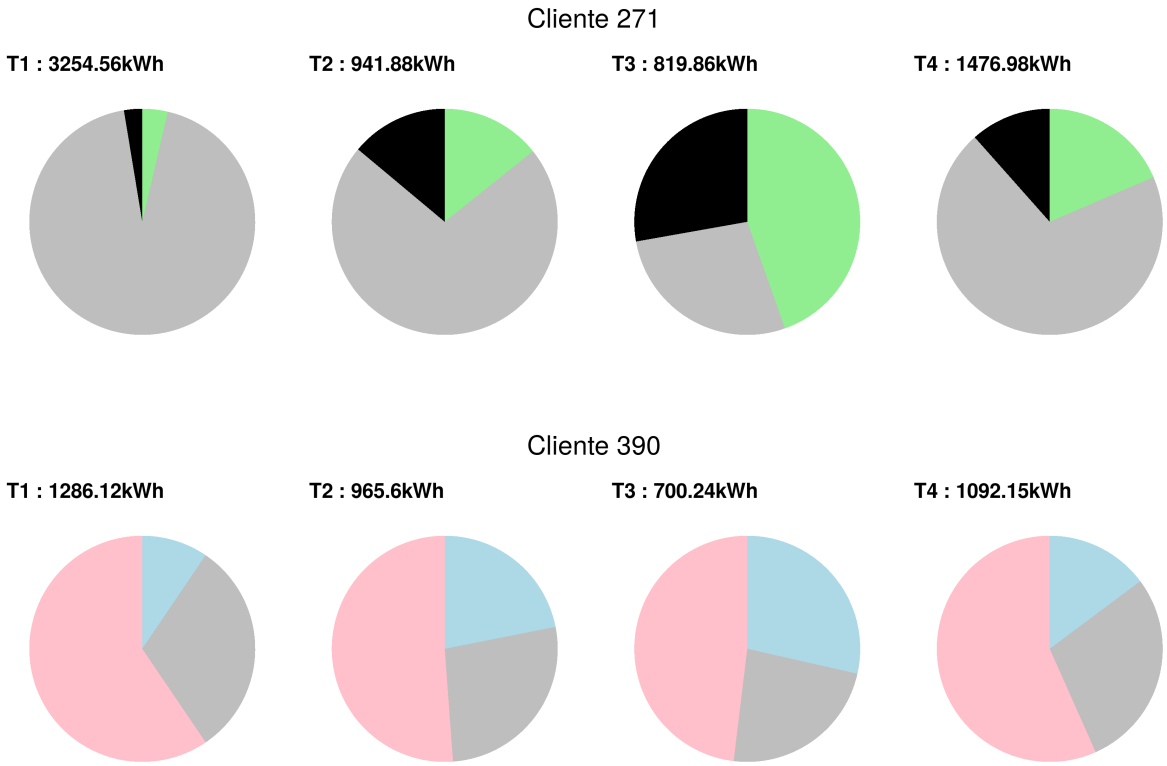
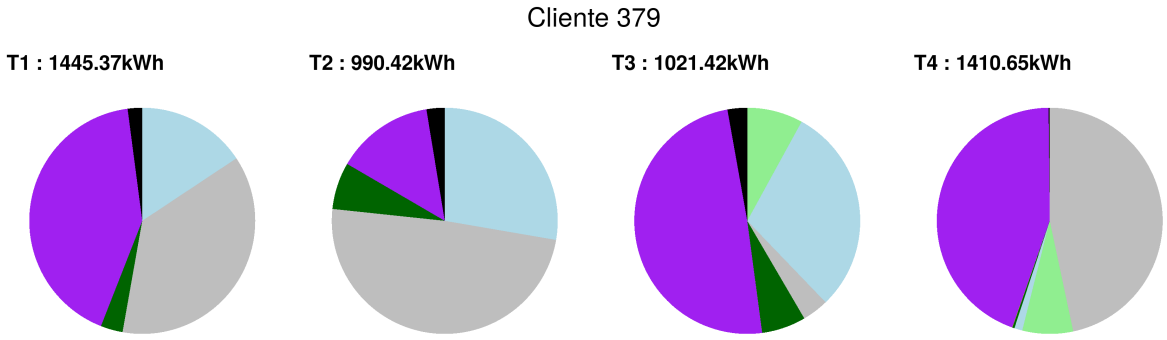


Figura 3.3: Gráficos circulares representativos dos consumos das *plugs* dos clientes 271 e 390 face aos seus consumos globais, para o ano de 2017.

Outro caso comum é o cliente que tem duas *plugs*. A maioria destes clientes não tem uma grande proporção do consumo global coberta pelas *plugs*. No entanto, certos clientes, como o cliente 390 (Figura 3.3) têm uma boa proporção do consumo medido com poucas *plugs*, presumivelmente por priorizarem os aparelhos usados com grande regularidade (que é o caso dos sistemas de águas quentes sanitárias que são geralmente utilizados diariamente por todos os membros do agregado familiar e dos aparelhos de refrigeração que estão constantemente ligados). Do mesmo modo, clientes como o cliente 271 conseguem explicar grande parte do seu consumo nos meses em que menos consomem, deixando a sua acentuada sazonalidade por medir (normalmente atribuída a aparelhos de climatização ou termoacumuladores).



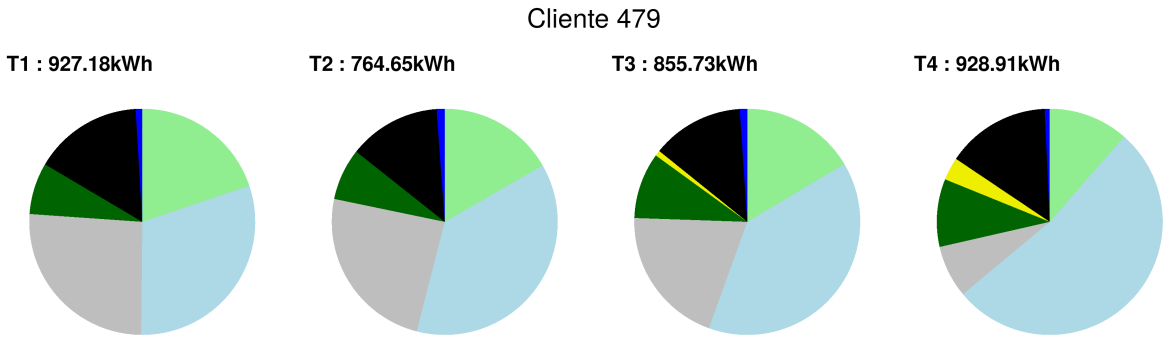


Figura 3.4: Gráficos circulares representativos dos consumos das *plugs* dos clientes 379 e 479 face aos seus consumos globais, para o ano de 2017.

Os clientes 379 e 479, cuja separação do consumo está apresentada na Figura 3.4 são raras ocorrências de clientes que possuem um número de *plugs* suficientemente grande para medir uma grande parte do seu consumo. Ambos adquiriram novas *plugs* em setembro, pelo que se verificam novas secções no gráfico a partir do terceiro trimestre.

A diversidade de clientes é aparente nestas representações em que se verifica que qualquer categoria pode ser responsável pela maioria do consumo do cliente.

3.2.2 Tabelas de frequências

Para ilustrar a informação presente na amostra, obteve-se a seguinte tabela de frequências (Tabela 3.3) que expõe o número de *plugs* associadas a cada categoria por cliente. Note-se que, num dos processos de seleção de clientes, se exigiu que estes tivessem *plugs* classificadas como frigoríficos, combinados ou máquinas de lavar roupa e loiça, causando um aumento nos números de clientes com pelo menos uma *plug* nas categorias de Refrigeração e Máquinas.

Tabela 3.3: Tabela de frequências referente ao número de *plugs* por cliente classificada com cada categoria.

Categoria/Subcategoria	Número de *plugs* associadas à categoria							
	0	1	2	3	4	5	6	7
Cozinha	408	86	14	0	1	0	0	0
Refrigeração	189	215	49	2	3	1	0	0
Máquinas	273	154	42	5	0	0	0	0
Multimédia	339	134	15	2	3	1	0	0
Informática	432	70	9	2	0	0	0	0
Iluminação	454	49	7	3	0	0	0	0
Aquecimento	412	56	12	2	0	3	1	1
Arrefecimento	489	12	5	2	1	1	0	0
Águas Quentes Sanitárias	474	40	3	2	0	0	0	0

Observando a Tabela 3.3, parece evidente que a distribuição do número de *plugs* associadas a cada categoria, está longe de ser semelhante à distribuição do número de aparelhos de cada categoria. Esta conclusão pode ser alcançada observando diversas categorias:

1. Espera-se que cada cliente tenha pelo menos um equipamento de refrigeração (frigorífico ou combinado) e que uma porção considerável tenha uma arca, no entanto grande parte dos clientes da amostra não tem nenhuma *plug* associada a aparelhos de refrigeração.

2. É seguro assumir que cada cliente terá pelo menos uma máquina de lavar roupa e que a grande maioria também possuirá uma máquina de lavar loiça. Ainda assim observam-se com maior frequência clientes com apenas uma ou nenhuma *plug* em máquinas do que clientes com duas.

3. Normalmente, o número de lâmpadas numa casa está na ordem das dezenas, no entanto o número máximo de *plugs* que se observam associadas à categoria de iluminação numa casa é 3.

Obteve-se uma tabela de frequências semelhante para os conjuntos de subcategorias selecionadas previamente para modelação (frigoríficos/combinados, máquinas de lavar roupa e máquinas de lavar loiça) – Tabela 3.4.

Tabela 3.4: Tabela de frequências referente às tipologia selecionadas para modelação.

	Número de *plugs* associadas à tipologia		
Categoria/Subcategoria	0	1	2
Frigorífico ou combinado	293	222	11
Máquina de lavar loiça	456	69	1
Máquina de lavar roupa	389	133	4

Embora à partida pareça haver um número grande de clientes para trabalhar cada uma das tipologias, é de salientar que parte destes clientes pode ter adquirido as *plugs* em questão após o início do ano de 2017, e consequentemente não ter histórico desse consumo parcial para o ano inteiro. Também pode acontecer que algumas das *plugs* em questão estejam mal classificadas.

3.2.3 Triagem de clientes e aparelhos por categoria

Uma vez que as *plugs* são classificadas pelos próprios clientes e que estas podem ser trocadas de aparelho em aparelho, foi feita uma triagem às *plugs* identificadas com as tipologias em estudo (Frigoríficos e Máquinas). Esta triagem tem como objetivo excluir os clientes que não têm os dados relativos a todo o ano de 2017 para os aparelhos em questão e os clientes cujos aparelhos estão categorizados erradamente. Deste modo os equipamentos restantes serão apenas os que contêm informação relevante para estimação dos consumos parciais mensais do ano de 2017.

Frigoríficos e Combinados Os aparelhos de refrigeração estão em utilização constantemente, ao contrário da maioria dos equipamentos que gastam mais energia quando estão a ser utilizados. Sendo assim, é possível validar um periférico como sendo um aparelho de refrigeração verificando que este consome constantemente, não havendo grandes picos de consumo ao longo do dia. Além disto, os frigoríficos têm, normalmente, potência elétrica compreendida entre 150 W e 400 W, pelo que se espera um consumo médio entre 0.0375 e 0.1 kWh para os intervalos de 15 minutos.

Para todos os clientes com *plugs* identificadas como frigoríficos ou frigoríficos combinados foi analisado o padrão de consumo dos aparelhos nos seguintes intervalos de 30 dias:

- de 1 a 30 de Janeiro;

- de 1 a 30 de Junho;
- de 2 a 31 de Dezembro.

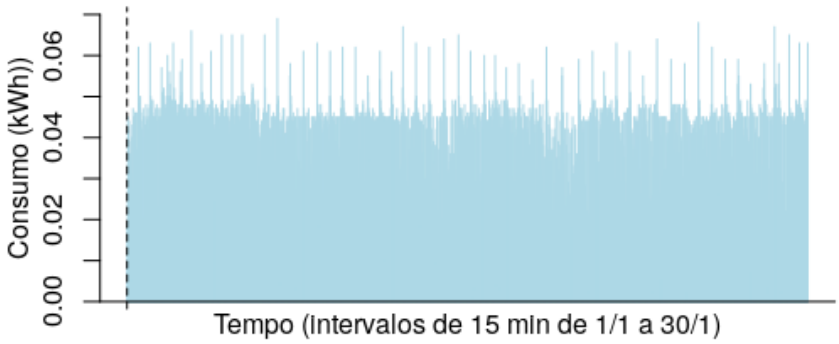


Figura 3.5: Consumo elétrico do frigorífico do cliente 1 nos primeiros 30 dias do ano.

O padrão de consumo apresentado na Figura 3.5 é o padrão expectável de um frigorífico, por não ter períodos sem consumo e apresentar valores de consumo plausíveis.

Figura 3.6: Consumo elétrico do frigorífico do cliente 1 no mês de Junho.

Na Figura 3.6 observa-se que o padrão de consumo apresenta intervalos sem consumo, seguidos de picos de consumos muito elevados. Este fenómeno deve-se a falhas de transmissão dos valores medidos pela *plug*, sendo os valores cujo envio falhou somados e enviados assim que possível (causando os picos de consumo). O padrão de consumo continua a ser plausível para um frigorífico e as falhas de transmissão pouco ou nada afetam os totais mensais. Após observar que a *plug* do cliente 1 apresenta um padrão semelhante para os últimos 30 dias do ano, considerou-se que esta terá estado realmente associada a um frigorífico no ano de 2017 inteiro.

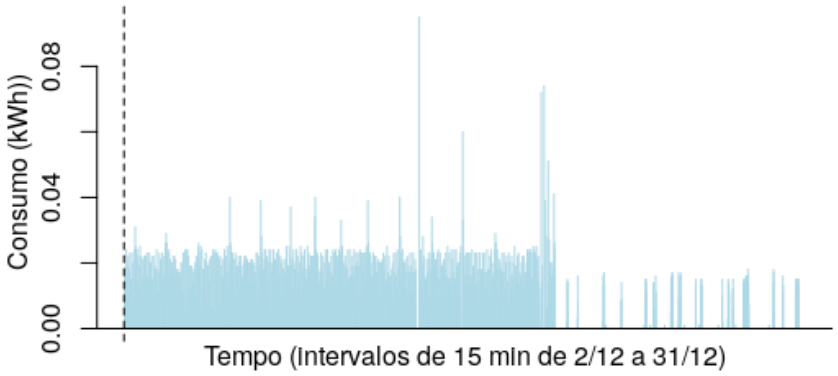


Figura 3.7: Consumo elétrico do frigorífico do cliente 6 nos últimos 30 dias do ano.

A *plug* que o cliente 6 identifica como frigorífico não foi considerada corretamente classificada, pois o padrão de consumo evidencia a troca da *plug* para um equipamento de uso esporádico no mês de Dezembro (Figura 3.7). Este cliente foi então excluído da amostra de clientes elegíveis para modelação do consumo parcial de frigoríficos.

Foram excluídos, também, os clientes cujas *plugs* do frigorífico não tinham histórico completo para o ano de 2017 e os clientes com mais do que um frigorífico (uma vez que se tratam de caso atípicos que poderiam influenciar bastante a estimativa do consumo parcial para os restantes).

De entre os 233 clientes com *plugs* categorizadas como frigoríficos, apenas 67 (47 com frigoríficos e 20 com frigoríficos combinados) foram considerados elegíveis para o processo de modelação. A grande maioria dos 166 clientes restantes foi descartada por ter falta de histórico de consumo, devido à aquisição das *plugs* em questão se ter dado após o início do ano.

Máquinas Analogamente ao processo de triagem das *plugs* classificadas como frigoríficos, foi efetuada a triagem às máquinas de lavar roupa e máquinas de lavar loiça. O comportamento esperado destes eletrodomésticos será um de utilização esporádica, ao contrário dos frigoríficos em que se esperava um consumo constante.

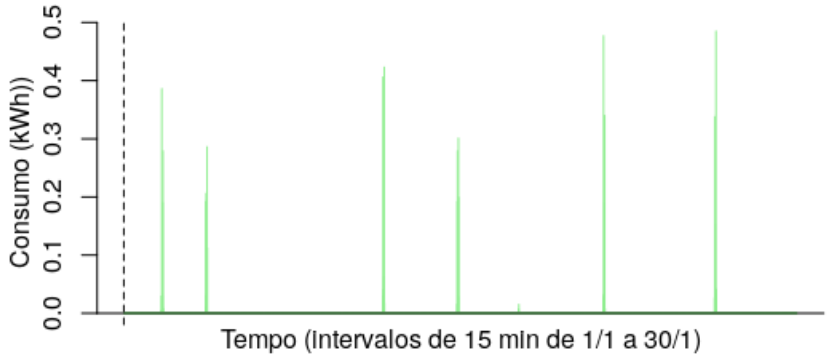


Figura 3.8: Consumo elétrico da máquina de lavar roupa do cliente 156 nos primeiros 30 dias do ano.

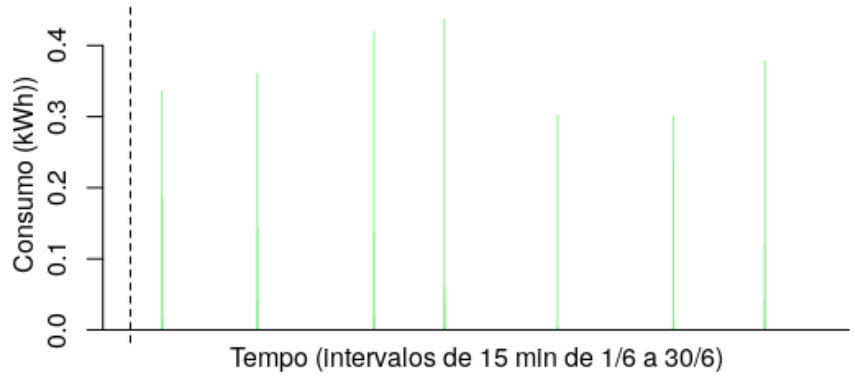


Figura 3.9: Consumo elétrico da máquina de lavar loiça do cliente 204 no mês de Junho.

Nas Figuras 3.8 e 3.9 observa-se o comportamento de máquinas. É de salientar que as máquinas da loiça são semelhantes às das máquinas da roupa em termos de potência média num intervalo de 15 minutos. De entre 137 clientes com máquinas de lavar roupa monitorizadas, apenas 30 foram considerados elegíveis. De entre 70 clientes com máquinas de lavar loiça monitorizadas, 18 foram considerados elegíveis para o processo de modelação. Novamente, a maioria dos clientes restantes foi descartada por não ter histórico de consumo para todo o ano de 2017.

3.2.4 *Box-plots*

O valor de consumo parcial mensal pode não ser sazonal para algumas tipologias, no entanto a maioria dos aparelhos apresentará variações de consumo ao longo do ano. Obtiveram-se *box-plots* paralelos para os consumos mensais que permitem observar a sazonalidade dos consumos das diferentes tipologias de aparelho.

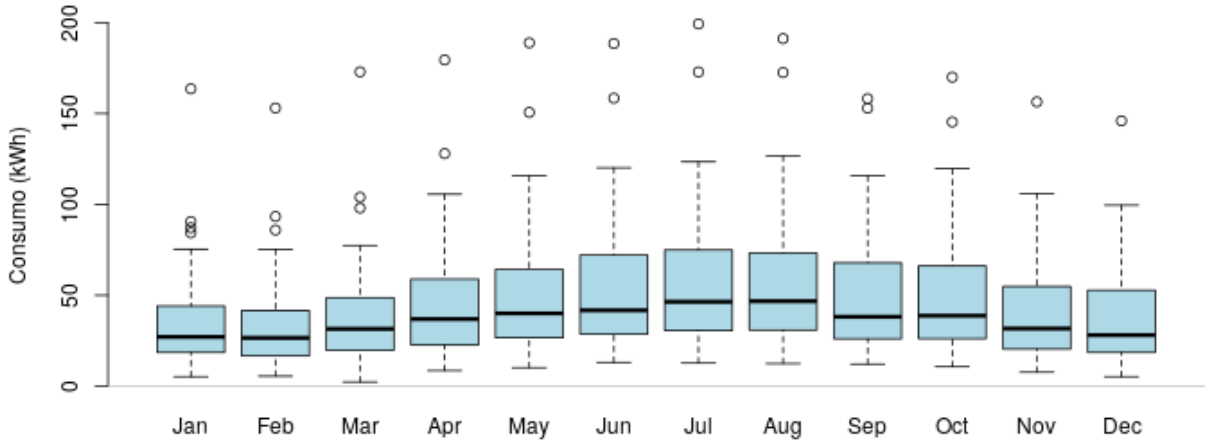


Figura 3.10: *Box-plots* paralelos dos consumos mensais de 68 frigoríficos e combinados.

A Figura 3.10 demonstra claramente a sazonalidade do consumo dos aparelhos de refrigeração como frigoríficos e combinados. O resultado era esperado, pois estes aparelhos consomem bastante mais energia nos meses de verão do que nos meses de inverno. A variação deve-se ao trabalho extra efetuado pelos aparelhos para manter uma temperatura interior baixa quando a temperatura exterior é mais alta.

Os comprimentos dos bigodes superiores dos diagramas são bastante grandes, que é sinal da variabilidade entre os frigoríficos de consumo mais elevado. A evolução da eficiência dos aparelhos de refrigeração nas últimas décadas leva a crer que estes aparelhos com nível de consumo elevado serão aparelhos mais antigos (menos eficientes).

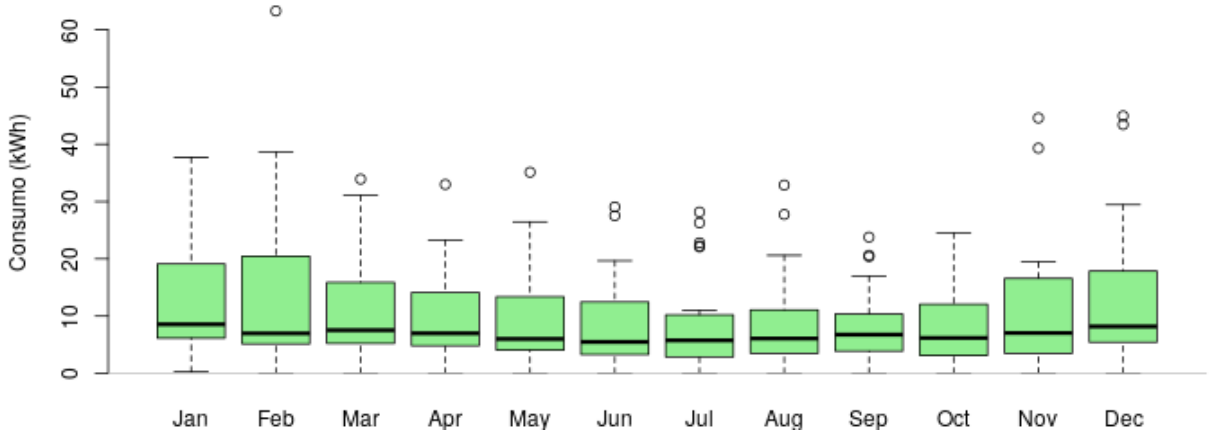


Figura 3.11: *Box-plots* paralelos dos consumos mensais de 31 máquinas de lavar roupa.

Inversamente ao sucedido com os frigoríficos e combinados, a Figura 3.11 leva a crer que existe maior consumo de energia ligado à lavagem de roupa no inverno que no verão. Uma possível explicação para tal seria o facto de a roupa usada por dia nos meses frios ser em maior peso e quantidade que nos meses quentes, gerando a necessidade de utilização da máquina mais frequentemente.

Neste caso, a variabilidade dos valores de consumo mais elevados (entre o terceiro quartil e o adjacente superior) é bastante diferente de mês para mês. Esta irregularidade dever-se-á provavelmente à esporadicidade do uso das máquinas e à reduzida dimensão da amostra. Os valores mais elevados dos consumos mensais dos frigoríficos dever-se-ão à fraca eficiência dos aparelhos, enquanto que os valores mais elevados do consumo das máquinas ocorrem devido a frequências atípicas de utilização do aparelho.

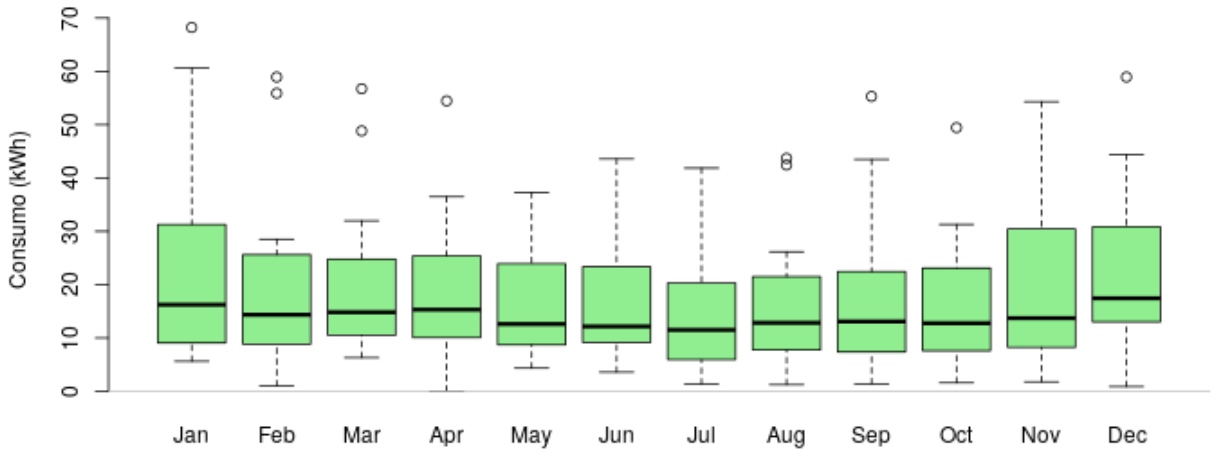
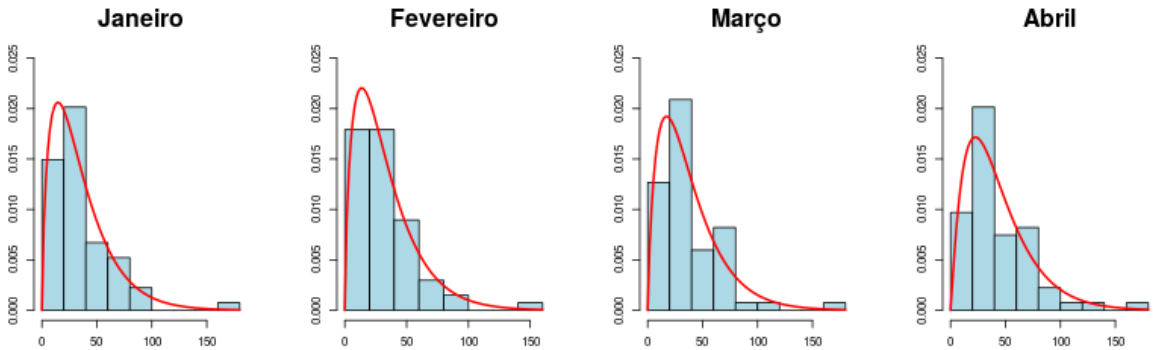


Figura 3.12: *Box-plots* paralelos dos consumos mensais de 19 máquinas de lavar loiça.

O padrão de sazonalidade do consumo elétrico em lavagem de loiça não é tão aparente (Figura 3.12), no entanto parece haver maior consumo no meses de inverno. A mediana do consumo de máquinas de lavar loiça por mês é bastante superior à mediana do consumo em lavagem de roupa, o que indica a utilização mais frequente no caso da lavagem da loiça (uma vez que se verificou na triagem de clientes que as potências médias num intervalo de 15 minutos das duas tipologias de aparelhos aparentam ser semelhantes).

3.2.5 Estudo da distribuição amostral

Para estudar a distribuição dos consumos parciais em cada mês, obtiveram-se os seguintes histogramas. Dada a forma dos histogramas e o facto de o consumo de energia não tomar valores negativos, começou-se por comparar os ditos histogramas com a curva da função de densidade de probabilidade da distribuição Gama.



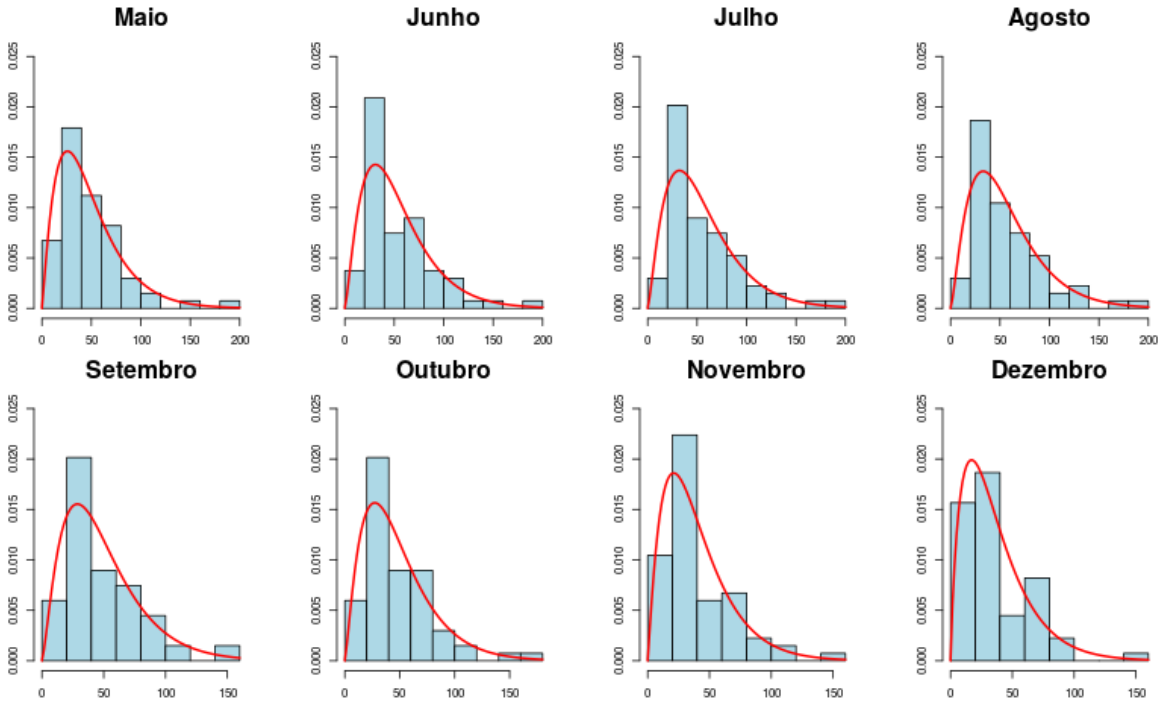


Figura 3.13: Histogramas de consumo de frigoríficos e combinados por mês. A vermelho: função de densidade de probabilidade da distribuição Gama (com parâmetros estimados por máxima verosimilhança).

Observa-se na Figura 3.13 que a distribuição Gama se parece ajustar de forma razoável ao consumo parcial dos frigoríficos na maioria dos meses. O facto de a distribuição dos dados ser eventualmente Gama é uma possível explicação para a observação dos candidatos a *outlier* nos *box-plots*, uma vez que esta distribuição tem a cauda direita relativamente pesada.

Para verificar se a distribuição Gama realmente se ajusta à amostra, foi aplicado o teste de ajustamento do Qui-Quadrado, sendo a hipótese a testar a seguinte:

$$H_0 : Consumo\ parcial\ mensal \sim Gama(\alpha, \beta) \tag{3.1}$$

O número de classes para cada amostra foi definido segundo a regra de Sturges. Os limites das classes foram escolhidos de forma a que estas cobrissem todo o domínio da distribuição subjacente à hipótese em teste e que as classes fossem intervalos equiprováveis sob a mesma distribuição. Os parâmetros α e β foram estimados através do Método da Máxima Verosimilhança, reduzindo em duas unidades os graus de liberdade da distribuição subjacente a H_0 da estatística de teste X^2.

$$X^2 = \sum_{j=1}^{k} \frac{(o_j - e_j)^2}{e_j} \overset{H_0}{\sim} \chi^2_{(k-1-2)} \tag{3.2}$$

k: "Número de classes."
o_j: "Frequência observada na classe j."
e_j: "Frequência esperada na classe j sob a distribuição subjacente a H_0."

Tabela 3.5: Valores observados da estatística de teste e do *p-value* relativos ao teste de ajustamento do Qui-Quadrado para cada mês – Frigoríficos/Combinados.

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
x^2	4,358	8,955	5,403	4,358	8,119	12,925	11,254	7,91	7,284	6,239	9,164	9,791
p-value	0,36	0,062	0,248	0,36	0,087	0,012	0,024	0,095	0,122	0,182	0,057	0,044

Considerando o nível de significância mais usual de $\alpha = 0{,}05$, apenas existe evidência para afirmar que os dados não provêm de uma distribuição Gama nos meses de junho, julho e dezembro. Ainda assim a hipótese nula nestes meses não é rejeitada para todos os níveis de significância usuais ($p\text{-}value > 0{,}01$).

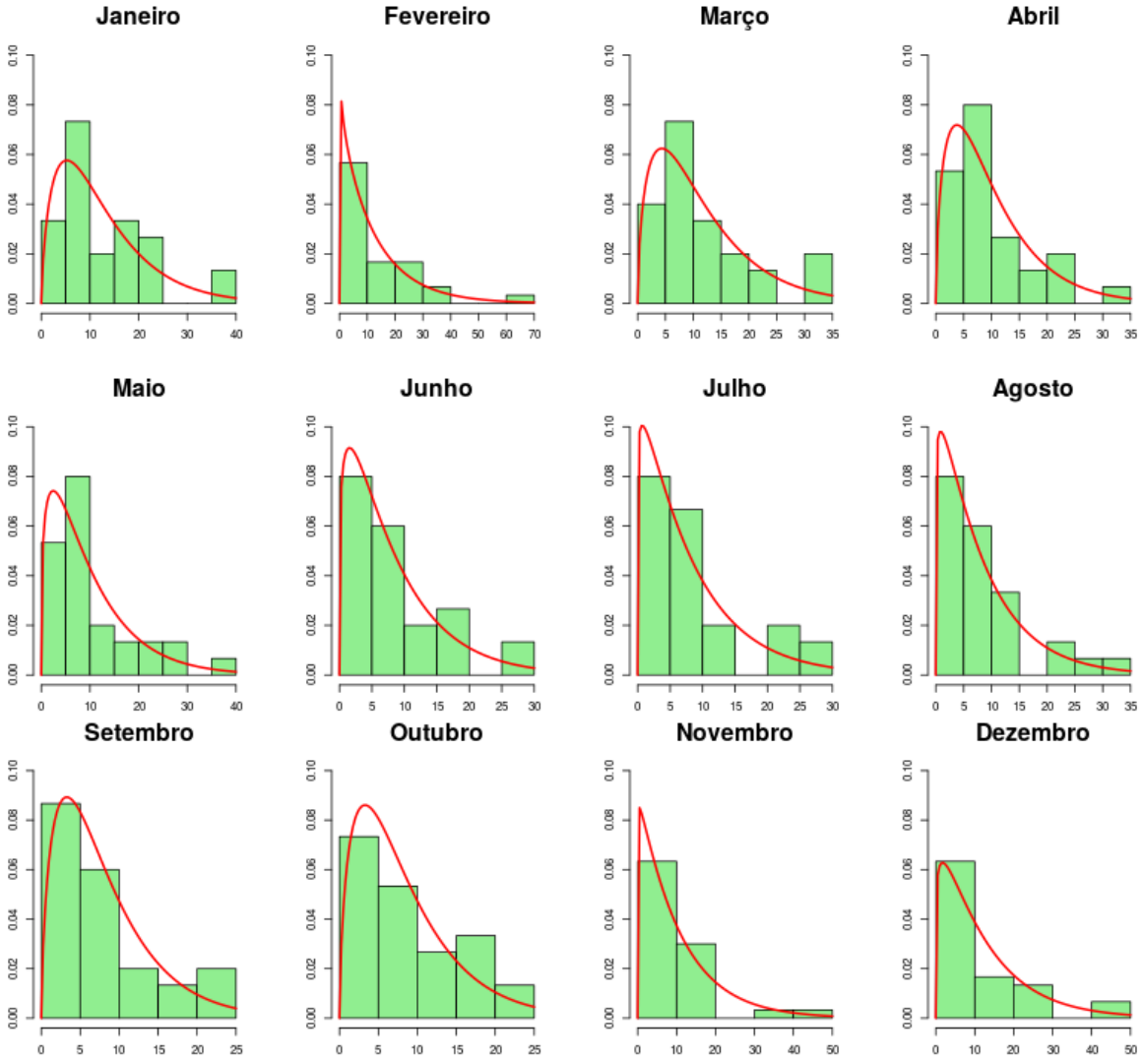


Figura 3.14: Histogramas de consumo de máquinas de lavar roupa por mês. A vermelho: função de densidade de probabilidade da distribuição Gama (com parâmetros estimados por máxima verosimilhança).

Relativamente às máquinas de lavar roupa, observa-se a semelhança entre os histogramas e a função de densidade de probabilidade da distribuição Gama nos gráficos da Figura 3.14, tal como no caso dos frigoríficos/combinados.

Novamente, foi aplicado o teste de ajustamento do Qui-Quadrado para verificar a qualidade de ajustamento da distribuição Gama a cada amostra mensal. Observando os resultados do teste de hipóteses na Tabela 3.6, não existe razão para afirmar que os dados não provêm de uma população com distribuição probabilística Gama em qualquer um dos doze meses, considerando o nível de significância $\alpha = 0{,}05$.

Tabela 3.6: Valores observados da estatística de teste e do *p-value* relativos ao teste de ajustamento do Qui-Quadrado para cada mês – Máquinas de Lavar Roupa.

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
x^2	4,667	4,333	5,667	4	5	0,667	1	1,667	1,667	0,667	2,333	1
p-value	0,097	0,115	0,059	0,135	0,082	0,717	0,607	0,435	0,435	0,717	0,311	0,607

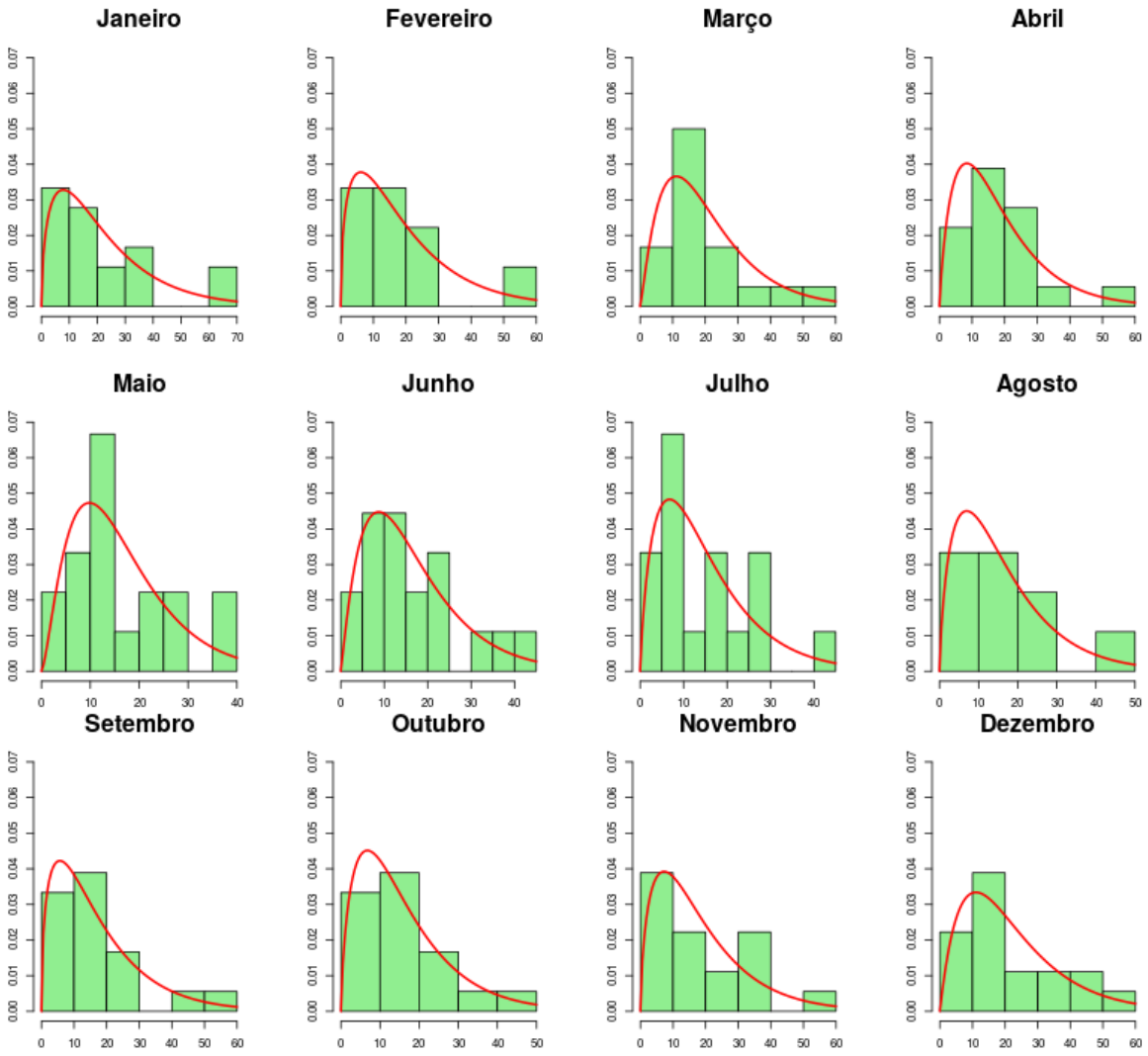


Figura 3.15: Histogramas de consumo de máquinas de lavar loiça por mês. A vermelho: função de densidade de probabilidade da distribuição Gama (com parâmetros estimados por máxima verosimilhança).

A qualidade do ajustamento da distribuição Gama para os dados de consumo das máquinas de lavar loiça parece ser idêntica à dos restantes tipos de eletrodomésticos, comparando os histogramas da Figura 3.15 com os apresentados anteriormente. No entanto, esta qualidade de ajustamento não foi testada, uma vez que a distribuição da estatística de teste do teste do Qui-Quadrado é assimptótica (aproximada através do teorema de De Moivre - Laplace) e que 18 seria um número de observações insuficiente para considerar tal aproximação.

4. Estimação de consumos de frigoríficos e máquinas

4.1 Variáveis independentes

Para a construção de um algoritmo preditivo para os consumos parciais do cliente, foi necessário utilizar a informação disponível para todos os clientes como ponto de partida para os modelos integrantes do algoritmo.

A construção de covariáveis que sejam realmente explicativas da variável resposta constitui um passo importante na obtenção de um modelo estatístico preciso [22]. Construiu-se um conjunto de 97 variáveis, cujos valores foram recolhidos para cada cliente.

De entre as 97 variáveis, 39 incluem os consumos globais mensais dos clientes, as médias de temperatura e outros valores calculados a partir dos já referidos. Espera-se que estas variáveis contenham informação sobre o nível de consumo do cliente e variação do consumo ao longo do ano. Estas variáveis estão descritas na Tabela 4.1.

Tabela 4.1: Covariáveis derivadas de dados mensais.

Variável	Descrição/ forma de cálculo
Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec	Consumo global do cliente para cada um dos 12 meses de 2017.
Min, Max, Var, Amp	Minímo, máximo, variância e amplitude dos consumos globais mensais de 2017.
Minx	Min/Max
Ampx	Amp/Max
Tot	Consumo global total de 2017.
Inv, Ver	Consumo global total nos meses de verão e consumo global total nos meses de inverno.
Dinver	Inv-Ver
MKurt, MSkew	Valores de curtose e achatamento para os consumos globais mensais.
T1, T2, ..., T12	Média de temperatura na zona geográfica do cliente para cada mês.
Cortemp	Coeficiente de correlação entre os consumos globais mensais e médias de temperatura mensais.
MQ25, MQ75	Quantis amostrais de probabilidades 0.25 e 0.75 para os consumos globais mensais.

As variáveis descritas na Tabela 4.2 foram obtidas com base em dados de consumo global diário. O objectivo é que estas capturem informação sobre a variação do consumo ao longo da semana (principalmente o comportamento do cliente em dias de semana e dias de fim-de-semana).

Tabela 4.2: Covariáveis derivadas de dados de consumo global diário.

Variável	Descrição/ forma de cálculo
AC1, AC2, ..., AC30	Valores dos coeficientes da função de autocorrelação dos consumos globais diários com até 30 dias de intervalo.
Wday	Média do consumo global em dias de semana.
Wend	Média do consumo global em dias de fim-de-semana.
Wincr	Aumento percentual do consumo médio em dias de fim-de-semana em relação ao consumo médio em dias de semana.
Wfraction	Fração do consumo total anual que se deu em dias de fim-de-semana.

Na Tabela 4.3, estão descritas as variáveis independentes derivadas dos valores de consumo global em intervalos de 15 minutos. Com estas variáveis pretende-se:

- identificar e comparar os níveis de consumo nas diferentes alturas do dia.
- obter informação sobre a *base load* do cliente, ou seja, o consumo constante da casa (principalmente composto por aparelhos de refrigeração e *stand by*). A variável Media4h foi incluída precisamente por ser a hora de menor consumo, na esperança de ser um bom indicador da *base load* (tal como os quantis amostrais de probabilidade baixa).
- incluir informação sobre a potência dos aparelhos de alto consumo através da análise da diferença entre observações consecutivas.
- incluir outras características amostrais utilizadas frequentemente na construção de covariáveis como a curtose e achatamento.

Tabela 4.3: Covariáveis derivadas de dados de consumo global (medido em intervalos de 15 minutos).

Variável	Descrição/ forma de cálculo
Q02, Q05, Q08, Q25, Q50, Q75, Q92, Q95, Q98	Quantis amostrais de probabilidades 0.02, 0.05, 0.08, 0.25, 0.5, 0.75, 0.92, 0.95, 0.98 para todas as observações do consumo global do cliente no ano de 2017 (com medições de 15 em 15 minutos).
Kurt, Skew	Valores de curtose e achatamento para todas as observações do consumo global do cliente no ano de 2017 (com medições de 15 em 15 minutos).
SADif	Soma das diferenças absolutas entre cada observação e a observação anterior para todas as observações do consumo global do cliente no ano de 2017 (com medições de 15 em 15 minutos).
Stepbin1	Proporção de observações nas quais se observa potência média para o intervalo de 15 minutos inferior a 500W.
Stepbin2	Proporção de observações nas quais se observa potência média para o intervalo de 15 minutos igual ou superior 500W mas inferior a 1000W.
Stepbin3	Proporção de observações nas quais se observa potência média para o intervalo de 15 minutos igual ou superior a 1000W.
Centrebin1	Média das observações nas quais se observa potência média para o intervalo de 15 minutos inferior a 500W.
Centrebin2	Média das observações nas quais se observa potência média para o intervalo de 15 minutos igual ou superior 500W mas inferior a 1000W.
Centrebin3	Média das observações nas quais se observa potência média para o intervalo de 15 minutos igual ou superior a 1000W.
Rquant	(Q02+Q05+Q08)/(Q92+Q95+Q98)
Media4h	Consumo global médio por dia entre as 4h e as 5h da madrugada.
Dawn, Morning, Afternoon, Evening	Consumo global médio por dia nas horas da madrugada, nas horas da manhã, nas horas da tarde e nas horas da noite.

Uma vez que o número de clientes aprovados pela triagem de equipamentos em cada categoria é inferior ao número de variáveis independentes, foi efetuada uma análise em componentes principais para reduzir o número de covariáveis. Assim, obtém-se um novo conjunto de covariáveis não correlacionadas entre si, eliminando o problema da multicolinearidade certamente presente no conjunto de variáveis originalmente apresentado (uma vez que grande parte das variáveis são relacionadas entre si).

Como *input* para o algoritmo de previsão, foram usadas 6 componentes principais, uma vez que a complexidade do algoritmo aumenta bastante a partir desse número e que uma das categorias em estudo (máquinas de lavar loiça) conta com apenas 18 observações. Os resultados foram estudados para duas formas de retenção de componentes principais:

- Seleção das 6 primeiras componentes principais (77.6% da variabilidade total da amostra inicial)
- Seleção das 6 componentes principais mais correlacionadas com o consumo parcial anual da categoria em questão

4.2 Estrutura dos algoritmos preditivos

Os algoritmos utilizados para a estimação dos consumos parciais associados a frigoríficos e máquinas seguem uma estrutura de *Ensemble Learning* com algoritmos de primeiro nível e algoritmos de segundo nível (meta algoritmos). Nesta estrutura, as previsões dos algoritmos de primeiro nível são obtidas através de uma validação cruzada *5-fold*, tal como explicado em [23].

Com o objetivo de combinar os pontos fortes de várias metodologias, foram selecionados alguns tipos de algoritmo bastante distintos. Abaixo está a lista dos algoritmos de primeiro nível utilizados:

- **Média do conjunto de treino.** Consiste em utilizar as médias das variáveis resposta para todas as observações disponíveis (conjunto de treino), como previsões para observações futuras. Mais do que algoritmo de primeiro nível, a média do conjunto de treino serve de referência para análise do desempenho dos restantes algoritmos.
- **Agrupamento pelos vizinhos mais próximos.** Utiliza a média dos valores observados das variáveis resposta para os k clientes "mais próximos"para previsão sobre o cliente em questão. Para k entre 1 e 7, o algoritmo faz seleção de covariáveis *backward stepwise*, selecionando sempre a variável cuja inclusão resulta no maior aumento da precisão do algoritmo (*Energy Accuracy*) dentro do conjunto de treino. Seleciona-se o conjunto de variáveis e o número de vizinhos (k) que produzem a maior precisão de entre todas as iterações do algoritmo. O algoritmo foi escrito em R para seguir um procedimento semelhante ao de [5].
- **Rede neuronal.** Uma rede neuronal com duas camadas ocultas com 5 e 3 vértices, respetivamente. Qualquer aumento no número de vértices, impacta bastante o tempo de execução. Foi utilizada a função *neuralnet* pertencente ao pacote *MASS* do R. O algoritmo utilizado para o ajustamento do modelo foi *RPROP*, uma variante do algoritmo *Backpropagation* [24].
- **Modelo linear.** Um modelo linear múltiplo de resposta multivariada. Foi utilizada a função *lm* pertencente ao pacote *stats* do R. A transformação Box-Cox foi aplicada

automaticamente a todas as variáveis resposta considerando a estimativa de máxima verosimilhança do parâmetro λ. Uma vez que as variáveis de interesse têm suporte positivo, todas as previsões negativas foram substituídas pelo valor 0.

- **Modelo linear generalizado.** Um modelo de regressão gama múltiplo, com resposta univariada (aplicado separadamente a cada mês do ano). A escolha deste modelo deve-se ao ajustamento relativamente bom da distribuição Gama aos dados de consumo parcial de cada mês. Foi utilizada a função glm pertencente ao pacote *stats* do R, usando o logaritmo como função de ligação. Para esta classe de modelos, foram utilizadas apenas 3 covariáveis (as três componentes principais com maior variância ou as três mais correlacionadas com o consumo parcial anual), devido a problemas de convergência do algoritmo.

- **Modelo de regressão robusta** Um modelo de regressão robusta múltipla e de resposta univariada baseado em mínimos quadrados aparados. Foi utilizada a função *lqs* do pacote *MASS* do R. Uma vez que as variáveis de interesse têm suporte positivo, todas as previsões negativas foram substituídas pelo valor 0.

- ***Gradient Boosting***. *Gradient Boosting* aplicado a modelos lineares univariados. É o modelo de eleição nas competições de modelação *online*. Foi utilizado o método *xgbLinear* do pacote *xgboost* do R. O pacote *caret* do R foi utilizado para estimar os hiperparâmetros m e η do modelo através de validação cruzada *5-fold*. Esta validação cruzada impacta drasticamente o tempo de execução do algoritmo.

Tal como descrito na Secção 2.8.3, as covariáveis dos modelos/algoritmos de segundo nível incluem as variáveis independentes iniciais (estimativas das componentes principais) e as estimativas das variáveis resposta dos algoritmos de primeiro nível. Foram selecionados vários algoritmos de segundo nível, não com o objetivo de combinar num algoritmo de terceiro nível mas para efeito de comparação.

- **Média das estimativas.** Ignorando os valores das estimativas das componentes principais, a estimativa final deste método é a médias das estimativas de primeiro nível.

- **Média ponderada das estimativas.** Foi aplicado um modelo de classificação (Floresta Aleatória) para prever qual seria a metodologia com menor erro absoluto para o caso específico de cada cliente. Este modelo recebe as estimativas das componentes principais e as previsões dos modelos de primeiro nível e tem como *output* a probabilidade estimada de cada modelo ter a melhor estimativa. Estas probabilidades servem de coeficientes de ponderação para as estimativas de primeiro nível no cálculo da estimativa final. Foi utilizada a função *randomForest* do pacote *randomForest* do R.

- **Seleção de uma estimativa de primeiro nível.** Seleciona-se, como estimativa final, a estimativa do modelo cuja probabilidade de ter menor erro absoluto estimada pela floresta aleatória é maior.

- **Modelo linear.** Modelo de segundo nível semelhante ao modelo linear de primeiro nível, acrescentando às variáveis independentes as estimativas de primeiro nível.

- **Rede neuronal.** Modelo de segundo nível semelhante à rede neuronal de primeiro nível, acrescentando às variáveis independentes as estimativas de primeiro nível e com resposta univariada (uma rede para cada mês).

Parte do código que constitui este processo está no anexo do relatório: a interface do processo no Apêndice A e o corpo do processo preditivo em si no Apêndice B.

Além dos algoritmos expostos, foi feita uma tentativa de implementação do algoritmo *Discriminative Disaggregation Sparse Coding* de Kolter et al. [4]. Este algoritmo revelou-se demasiado exigente computacionalmente para os processadores utilizados no estágio, o que impossibilitou a sua aplicação.

4.3 Resultados

Obtiveram-se, para os três tipos de equipamento selecionados (frigoríficos, máquinas de lavar roupa e máquinas de lavar loiça), previsões por cada um dos doze algoritmos mencionados. As previsões para cada cliente foram obtidas através de uma validação cruzada *Leave-One-Out*, utilizando os dados dos restantes clientes.

Para cada algoritmo foram comparados os valores de duas medidas de erro das previsões (*RMSE* e *MAE*) e o valor da medida de precisão sugerida por Batra et al. [5] (*Energy Accuracy*). Além disto, foram analisados os diagramas de dispersão entre valores observados e valores preditos pelos algoritmos com melhor desempenho.

4.3.1 Frigoríficos e Combinados

Obtiveram-se previsões de validação cruzada *Leave-One-Out* para os consumos parciais de frigoríficos e combinados de cada cliente. Foram utilizados dois procedimentos de seleção de covariáveis: seleção das seis componentes principais com maior variância e seleção das seis componentes principais mais correlacionadas com o consumo anual parcial de frigoríficos e combinados do conjunto de treino como variáveis independentes. Observando as medidas de diagnóstico dos erros de previsão para os dois conjuntos de covariáveis, concluiu-se que as previsões relativas ao segundo conjunto de covariáveis eram de qualidade bastante inferior às alcançadas partindo das primeiras componentes principais (maiores erros e menor precisão para todos os algoritmos). Na Tabela 4.4 apresentam-se as medidas de diagnóstico para o procedimento que utiliza as seis primeiras componentes principais como covariáveis. A amostra é composta por 67 clientes, 47 utilizadores de frigoríficos e 20 utilizadores de combinados.

Tabela 4.4: Precisão média e erros observados para cada algoritmo na previsão do consumo parcial de frigoríficos e frigoríficos combinados (utilizando as primeiras 6 componentes principais como covariáveis).

	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	31,02	23,41	45,59%
Vizinhos mais Próximos	37,48	30,04	35,76%
Rede Neuronal	45,28	30,51	46,76%
Modelo Linear	31,49	21,78	54,07%
Modelo Linear Generalizado	31,75	23,79	46,52%
Modelo de Regressão Robusta	32,90	21,99	55,66%
Gradient Boosting	38,35	28,12	46,08%
Média das Estimativas	31,14	22,79	50,23%
Média Ponderada das Estimativas	31,33	22,44	51,51%
Seleção de uma Estimativa	35,83	25,40	49,02%
Modelo Linear (2º Nível)	32,47	22,54	52,09%
Rede Neuronal (2º Nível)	39,00	27,22	47,40%

À partida, é sinal de mau desempenho dos restantes algoritmos que o *RMSE* mais baixo observado seja o da média das observações da variável resposta do conjunto de treino (o procedimento incluído como referência para comparação). Os altos valores de *RMSE* comparativamente a esta metodologia de referência sugerem que o conjunto de variáveis independentes não tenha uma relação suficientemente forte com as variáveis de interesse (consumos parciais mensais) ou que a amostra não tenha dimensão suficientemente grande para ajustar um bom modelo.

Os modelos de regressão linear têm a melhor *performance* em termos de *Energy Accuracy* e de *MAE*. As médias aritmética e ponderada das estimativas de primeiro nível têm um *RMSE* mais baixo que o dos modelos lineares. Selecionaram-se quatro modelos para comparar através de diagramas de dispersão (Figura 4.1) entre valores reais e valores preditos:

- O modelo linear e o modelo de regressão robusta, por apresentarem os melhores valores de precisão e erro absoluto;
- A média do conjunto de treino, que além de servir de referência, apresenta o erro quadrático médio mais baixo;
- A média das estimativas, que tem *RMSE* próximo do mais baixo observado, mantendo a precisão acima de 50%.

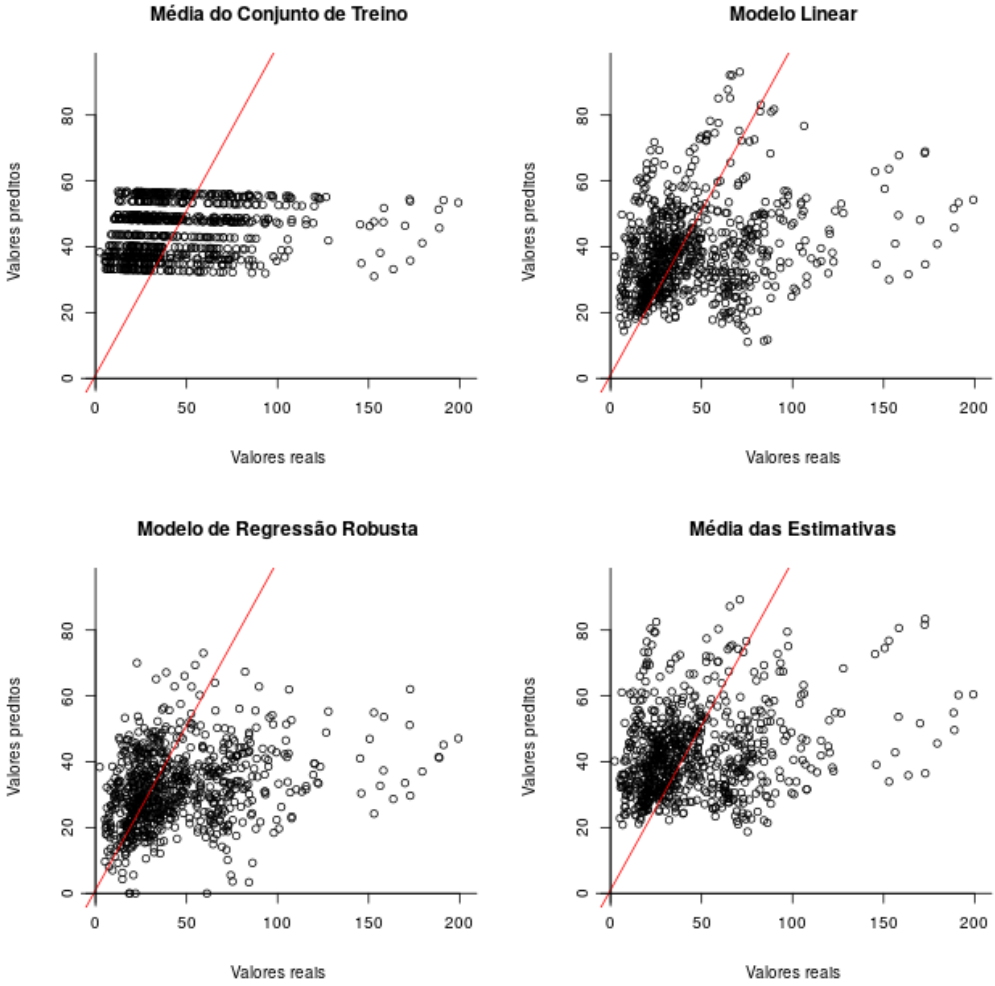


Figura 4.1: Diagramas de dispersão entre valores reais e preditos do consumo parcial pelos quatro algoritmos selecionados.

Em cada diagrama, a reta vermelha representa a bissetriz do primeiro quadrante do referencial cartesiano. Nestes diagramas cada cliente é representado por 12 observações (uma para cada mês).

Qualquer um dos quatro procedimentos selecionados tem erros grandes quando os valores reais são elevados. Em particular, o modelo linear parece ter um ajustamento razoável para valores reais do consumo parcial relativamente baixos, mas não tem capacidade de identificar as situações em que os valores desta variável são relativamente elevados. Esta falta de capacidade dos modelos leva a crer que a aplicação do algoritmo separadamente à subcategoria de frigoríficos e à subcategoria de combinados pode ser proveitosa uma vez que, como os combinados têm um compartimento para congelação de maior dimensão, se espera que estes sejam responsáveis pelos valores de consumo mais alto.

Outros procedimentos, como a rede neuronal, fazem previsões acima de 100 kWh, ao contrário dos modelos lineares. No entanto estes mantêm a inabilidade da identificação das situações em que o consumo parcial é realmente elevado (Figura 4.2).

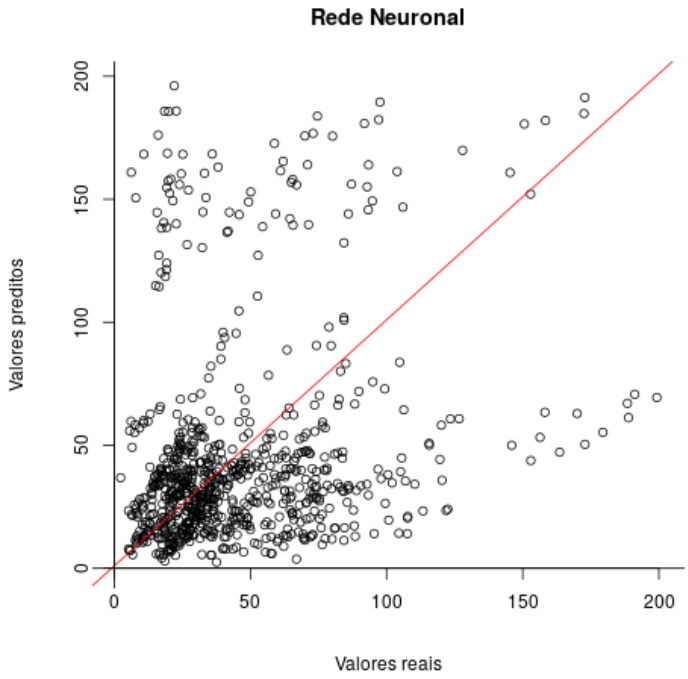


Figura 4.2: Diagrama de dispersão entre valores reais e preditos do consumo parcial pela rede neuronal.

4.3.2 Frigoríficos

O conjunto de algoritmos foi aplicado à amostra composta exclusivamente por clientes utilizadores de frigoríficos (47 clientes). Neste caso, ao contrário do sucedido com a amostra conjunta de clientes utilizadores de frigoríficos e combinados, os melhores resultados foram atingidos utilizando as componentes principais mais correlacionadas com o consumo parcial anual dos frigoríficos dos clientes. Os resultados deste procedimento foram avaliados através da Tabela 4.5 que contém os valores de *RMSE*, *MAE* e *Energy Accuracy* referentes a cada algoritmo.

Os valores dos erros absolutos e erros quadráticos são muito inferiores ao resultantes da aplicação dos algoritmos à amostra conjunta. O facto de os valores de *Energy Accuracy* serem mais baixos que nos resultados da experiência anterior (apesar da grande melhoria em termos

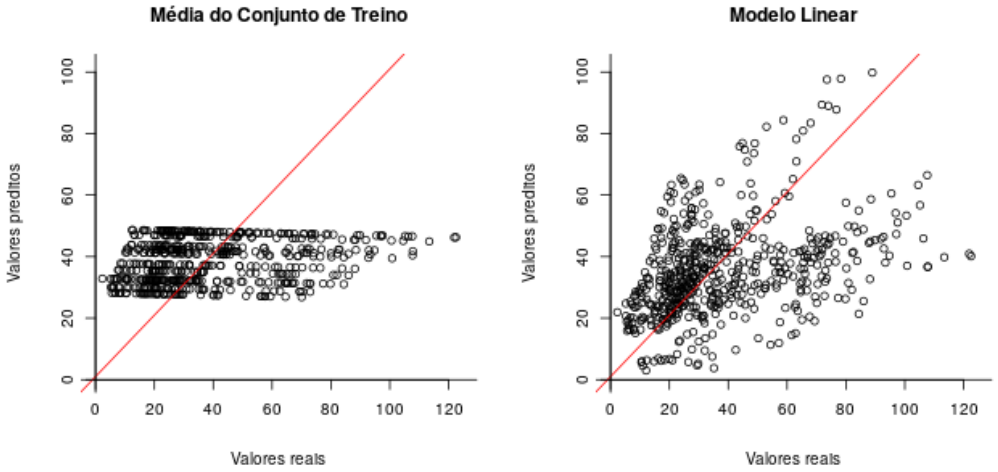
de *MAE* e *RMSE*) dever-se-á a esta ser uma medida de erros relativos e, consequentemente, ser mais exigente quando os valores reais das variáveis de interesse são mais baixos.

Tabela 4.5: Precisão média e erros observados na previsão do consumo parcial de frigoríficos (utilizando as 6 componentes principais mais correlacionadas com o consumo parcial anual como covariáveis).

	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	23,04	18,53	49,25%
Vizinhos mais Próximos	30,14	25,22	36,58%
Rede Neuronal	27,93	22,07	43,93%
Modelo Linear	22,75	17,57	53,46%
Modelo Linear Generalizado	23,58	18,48	51,21%
Modelo de Regressão Robusta	23,81	18,07	52,35%
Gradient Boosting	25,36	20,11	47,20%
Média das Estimativas	21,63	17,43	50,92%
Seleção de uma Estimativa	21,83	17,77	50,26%
Média Ponderada das Estimativas	25,69	20,61	46,68%
Modelo Linear (2º Nível)	23,06	17,82	51,98%
Rede Neuronal (2º Nível)	27,39	21,51	46,01%

Observam-se, na Figura 4.3, os diagramas de dispersão entre os valores reais e valores preditos de consumo parcial para quatro procedimentos:

- Modelo Linear, por ter o valor mais alto de precisão média (*Energy Accuracy*);
- Médias aritmética e ponderada das estimativas por apresentarem os valores mais reduzidos de *MAE* e *RMSE*;
- Média do Conjunto de Treino, como referência para comparação com os restantes algoritmos.



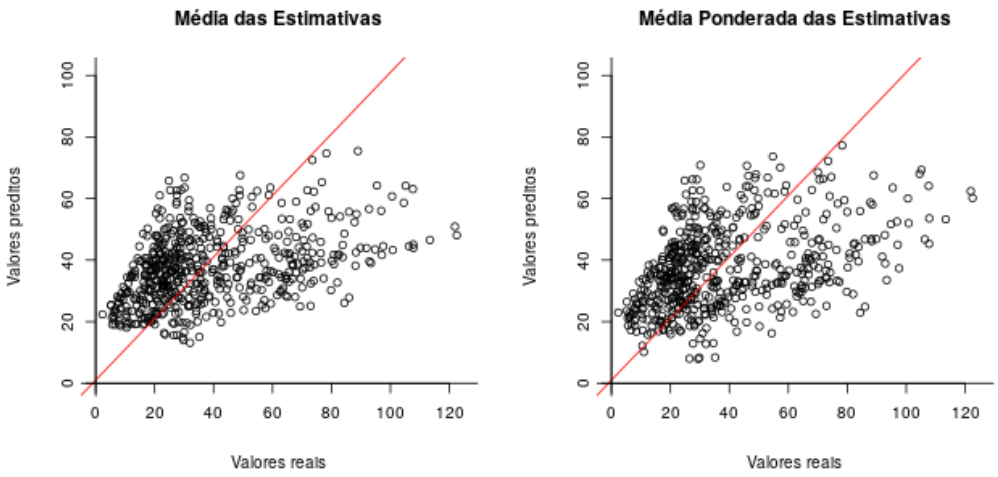


Figura 4.3: Diagramas de dispersão entre valores reais e preditos do consumo parcial pelos quatro algoritmos selecionados.

Observando a Figura 4.3, verifica-se que, apesar da exclusão dos clientes utilizadores de combinados, a incapacidade dos algoritmos de detetar as situações em que os valores de consumo parcial são mais elevados se mantém, apesar de menos acentuada. O melhor desempenho do modelo linear pode dever-se ao facto de a distribuição probabilística dos consumos parciais de frigoríficos ser aproximadamente Gama (visível na Figura 4.4), uma vez que se aplicou a transformação Box-Cox às variáveis resposta cada vez que se ajustou um modelo deste tipo.

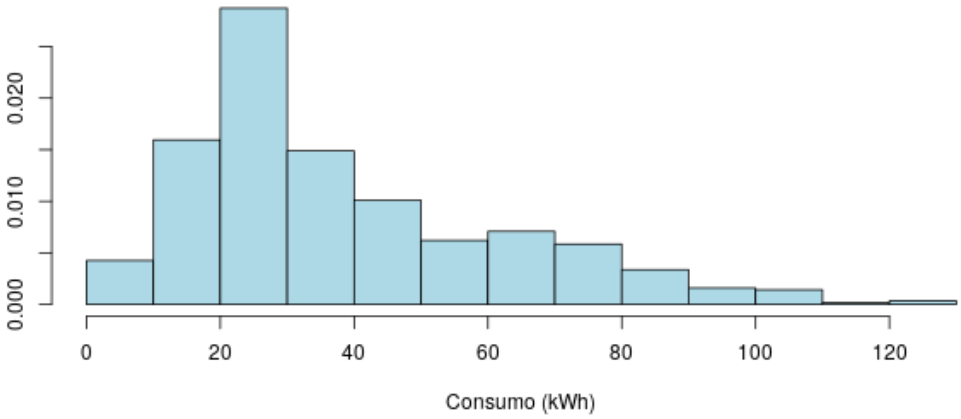


Figura 4.4: Histograma dos consumos parciais dos frigoríficos.

Apesar de à partida ser mais indicado para esta amostra, o modelo linear generalizado (modelo de regressão Gama) não parece ser uma melhor generalização para os dados em estudo (Figura 4.5).

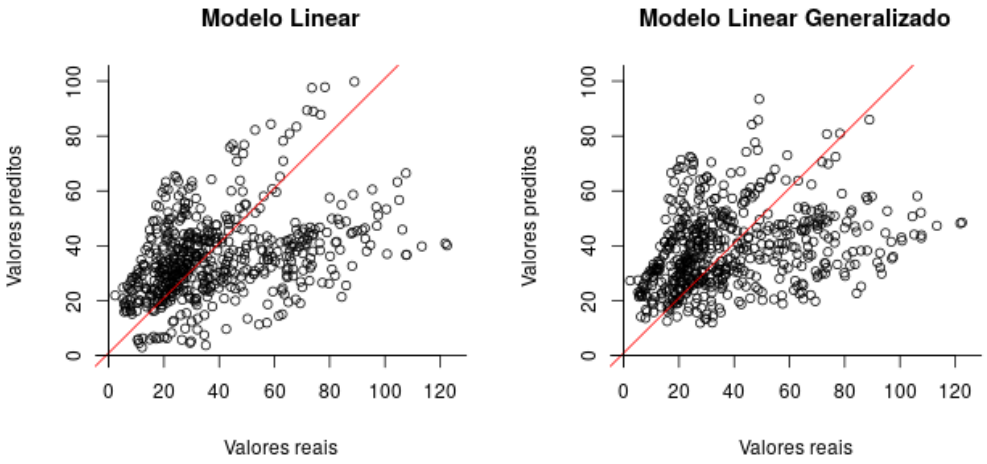


Figura 4.5: Diagrama de dispersão entre valores reais e preditos do consumo parcial pelos modelos linear e linear generalizado.

4.3.3 Combinados

Tal como foi aplicado aos clientes utilizadores de frigoríficos, o conjunto de algoritmos foi aplicado e avaliado para o conjunto de 20 clientes utilizadores de combinados. Novamente, a utilização das componentes principais mais correlacionadas com o consumo parcial anual teve melhores resultados pelo que se apresentam os valores observados das medidas de diagnóstico para este procedimento na Tabela 4.6.

Tabela 4.6: Precisão média e erros observados na previsão do consumo parcial de combinados (utilizando as 6 componentes principais mais correlacionadas com o consumo parcial anual como covariáveis).

	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	42,43	32,22	44,13%
Vizinhos mais Próximos	53,71	38,69	44,38%
Rede Neuronal	47,07	36,13	42,74%
Modelo Linear	38,35	27,96	51,08%
Modelo Linear Generalizado	39,09	26,26	55,58%
Modelo de Regressão Robusta	47,09	36,51	36,34%
Gradient Boosting	39,15	26,60	59,79%
Média das Estimativas	36,08	25,83	57,19%
Seleção de uma Estimativa	44,21	32,32	48,43%
Média Ponderada das Estimativas	36,94	26,79	55,41%
Modelo Linear (2º Nível)	46,62	35,78	39,73%
Rede Neuronal (2º Nível)	36,19	25,48	56,84%

Os valores de *Energy Accuracy* obtidos são melhores relativamente às experiências anteriores, possivelmente devido à maior tolerância desta medida quando os valores reais são elevados. Por outro lado, os valores de *RMSE* e *MAE* são relativamente elevados. Os três algoritmos que mais se destacam são:

- A média das estimativas de primeiro nível, por ter o menor valor de *RMSE*;
- A rede neuronal de segundo nível, por ter o menor valor de *MAE*;
- O modelo de *Gradient Boosting*, por ter o maior valor de *Energy Accuracy*.

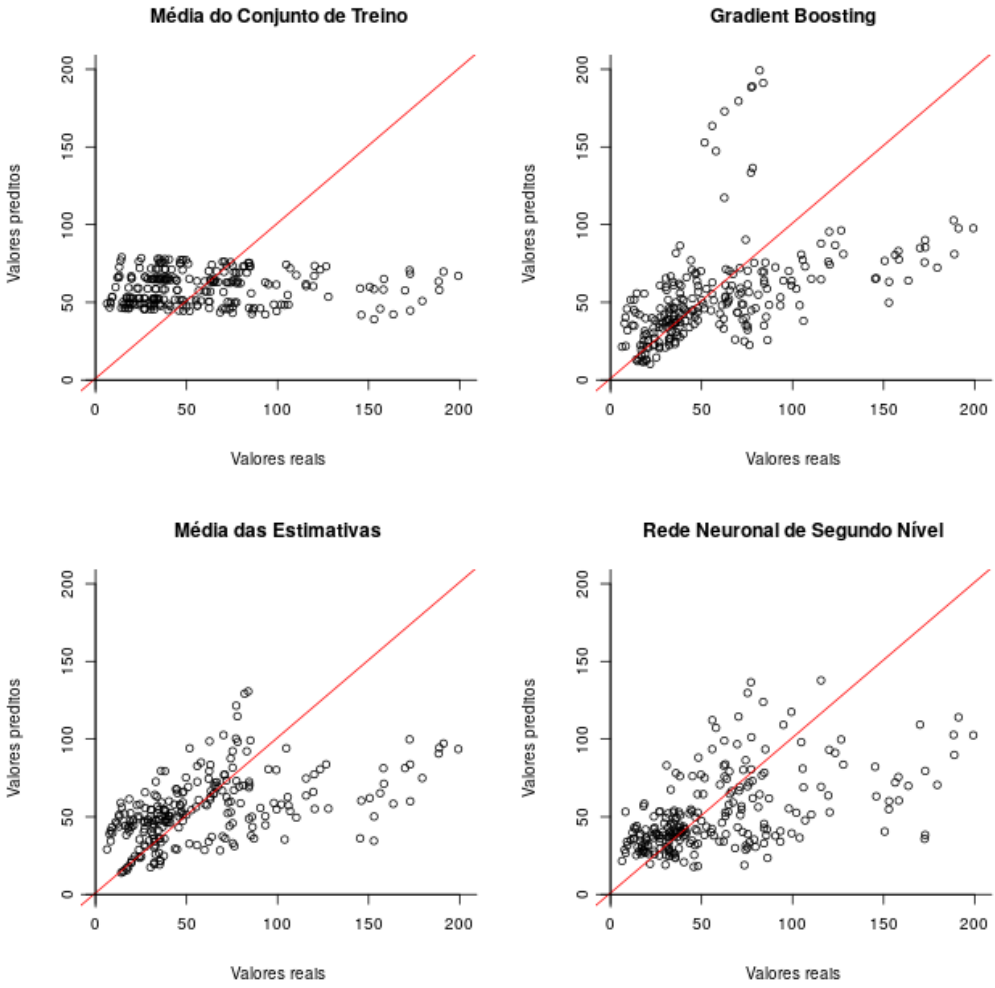


Figura 4.6: Diagramas de dispersão entre valores reais e preditos do consumo parcial pelos dois algoritmos selecionados.

Observando a Figura 4.6, as previsões da média das estimativas de primeiro nível e da rede neuronal de segundo nível parecem variar de acordo com a variação do consumo parcial. As previsões mais distantes dos valores reais são as que dizem respeito aos únicos dois clientes para os quais se observam valores de consumo parcial acima de 130, o que sugere que estes procedimentos poderiam ser capazes de produzir boas previsões caso se aumentasse a dimensão amostral.

4.3.4 Máquinas de Lavar Loiça

Os melhores resultados em termos de erro de previsão absolutos médio e erro de previsão quadrático médio no caso das máquinas de lavar loiça estão associados à média do conjunto de treino, utilizando ambas as metodologias de seleção das componentes principais. O único valor de *Energy Accuracy* mais alto que o observado na aplicação da média do conjunto de treino como preditor está associado à rede neuronal de primeiro nível (utilizando as primeiras 6 componentes principais como covariáveis). Ainda assim os valores de *RMSE* e *MAE* são bastante díspares entre estes dois procedimentos, como se pode observar na Tabela 4.7. O algoritmo cujos valores de *MAE* e *RMSE* mais se aproximam dos valores associados à média do conjunto de treino é a média ponderada das estimativas, que ainda assim tem resultados bastante piores.

Tabela 4.7: Precisão média e erros observados na previsão do consumo parcial de máquinas de lavar loiça (utilizando as primeiras 6 componentes principais como covariáveis).

	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	14,06	11,01	42,11%
Rede Neuronal	19,19	14,25	43,23%
Média Ponderada das Estimativas	16,45	13,29	35,75%

Ao observar a Figura 4.7, conclui-se que a rede neuronal não parece ter capacidade de explicar a variação das variáveis de interesse através desta amostra, uma vez que se observa uma grande proporção de previsões distantes do valor real. Esta conclusão seria de esperar apenas observando os valores das funções dos erros de previsão para as duas metodologias, uma vez que a média do conjunto de treino, sem qualquer tentativa de generalização dos dados consegue valores de *RMSE* e *MAE* muito inferiores aos da rede neuronal.

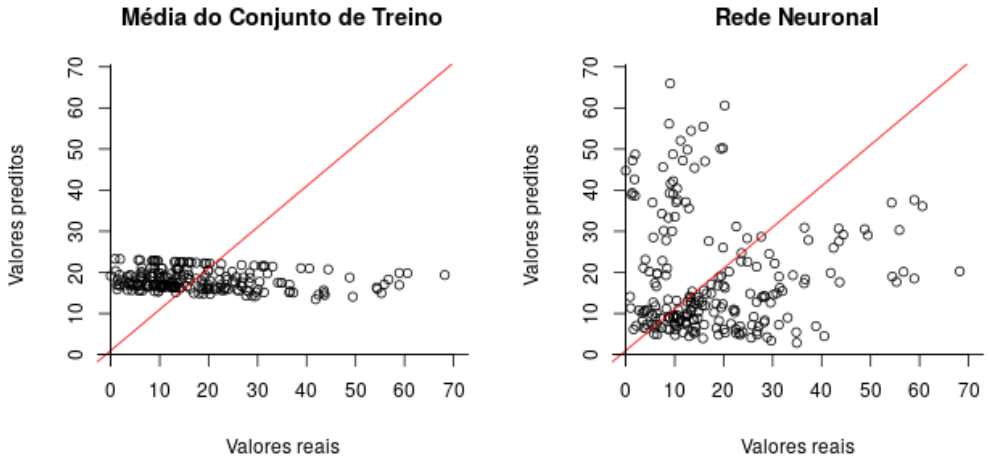


Figura 4.7: Diagramas de dispersão entre valores reais e preditos do consumo parcial pelos dois algoritmos selecionados.

As máquinas de lavar loiça tratam-se de um tipo de aparelhos cujo consumo depende do tipo de utilização que o cliente faz, ao contrário dos equipamentos de refrigeração que se encontram em utilização constante. Dada esta informação, colocou-se a possibilidade de o número de pessoas pertencentes ao agregado familiar do cliente ser uma variável importante para a previsão do consumo parcial das máquinas da loiça. Esta possibilidade não foi posta em prática pois da amostra inicial de 18 clientes com máquina de lavar loiça medida apenas se conhece a dimensão do agregado familiar para 12 clientes. O problema da aplicação do algoritmo preditivo a uma amostra desta dimensão é que o número de covariáveis (7 covariáveis: 6 componentes principais e dimensão do agregado familiar) se aproxima bastante da dimensão mínima dos conjuntos de treino na validação cruzada *5-fold* (8 clientes).

4.3.5 Máquinas de Lavar Roupa

Tal como sucedido com as máquinas de lavar loiça, os valores de *RMSE* e *MAE* mais baixos observados na previsão do consumo parcial das máquinas de lavar roupa são resultado da aplicação da média do conjunto de treino como algoritmo preditivo. No entanto, em contraste com a subclasse de aparelhos anterior, observam-se valores destas medidas diagnóstico bastante próximos aos associados à média do conjunto de treino nos outros algoritmos e vários destes têm uma precisão superior (utilizando as primeiras 6 componentes principais) (Tabela 4.8).

Tabela 4.8: Precisão média e erros observados na previsão do consumo parcial de máquinas de lavar roupa (utilizando as primeiras 6 componentes principais como covariáveis).

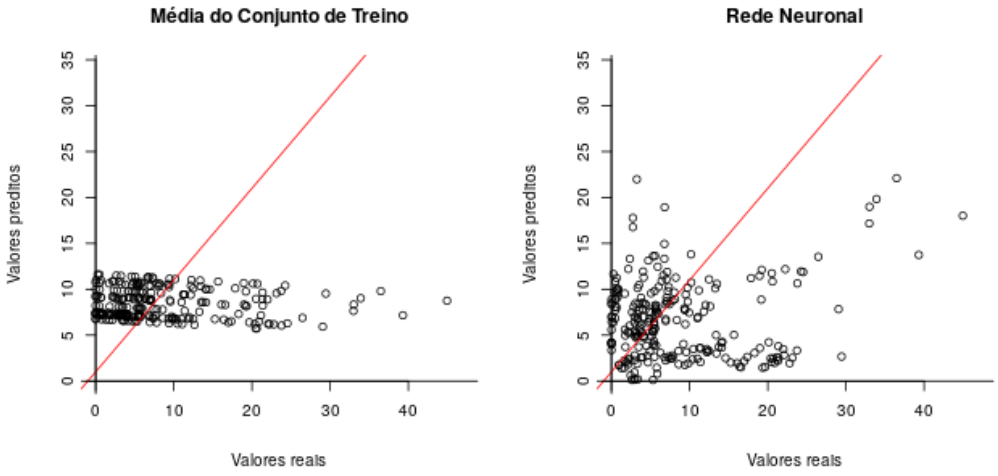
	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	9,31	7,10	35,24%
Vizinhos mais Próximos	10,63	7,63	36,67%
Rede Neuronal	15,97	11,58	30,20%
Modelo Linear	11,15	8,06	37,61%
Modelo Linear Generalizado	10,71	7,91	38,87%
Modelo de Regressão Robusta	12,19	8,76	28,19%
Gradient Boosting	11,36	8,77	31,90%
Média das Estimativas	9,83	7,36	37,47%
Seleção de uma Estimativa	11,92	8,97	31,77%
Média Ponderada das Estimativas	10,12	7,69	36,22%
Modelo Linear (2º Nível)	12,31	8,94	30,38%
Rede Neuronal (2º Nível)	13,18	9,96	31,32%

Novamente, colocou-se a possibilidade da inclusão da dimensão do agregado familiar do cliente como covariável. Neste caso, a amostra de 30 clientes fica reduzida aos 20 clientes para os quais esta informação está disponível. Utilizando a dimensão do agregado familiar em conjunto com as 6 componentes principais mais correlacionadas com o consumo parcial anual, obtiveram-se valores de *MAE* e *RMSE* inferiores ao da média do conjunto de treino para a rede neuronal. A precisão mais alta foi alcançada pela rede neuronal de segundo nível (Tabela 4.9).

Tabela 4.9: Precisão média e erros observados na previsão do consumo parcial de máquinas de lavar roupa (utilizando as 6 componentes principais mais correlacionadas com o consumo parcial anual e a dimensão do agregado familiar do cliente como covariáveis).

	RMSE	MAE	Energy Acc.
Média do Conjunto de Treino	8,19	6,21	34,15%
Rede Neuronal	7,51	5,72	36,89%
Rede Neuronal (2º Nível)	8,60	6,14	41,62%

Ainda que se tenham alcançado maiores precisões para as duas redes neuronais e menores valores de RMSE e MAE para a rede de primeiro nível utilizando a dimensão do agregado familiar, a análise dos diagramas de dispersão entre valores reais e valores preditos leva a crer que o modelo não capta a variabilidade das variáveis resposta (Figura 4.8).



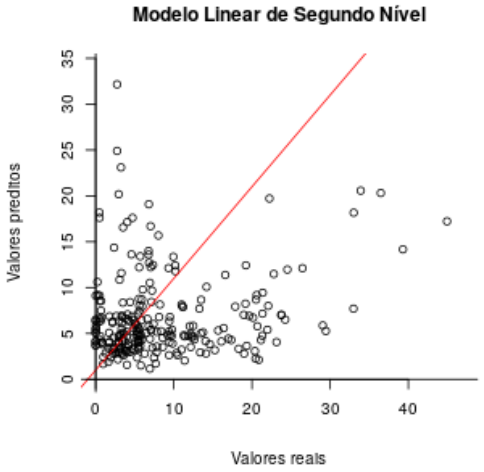


Figura 4.8: Diagramas de dispersão entre valores reais e preditos do consumo parcial pelos algoritmos selecionados.

5. Estimação de consumos de aquecimento ambiente

À parte do projeto de modelação dos consumos parciais de frigoríficos e máquinas, foi desenvolvida uma forma de previsão do consumo de aparelhos de aquecimento ambiente, com o objetivo de notificar o cliente se se observar que a estimativa do consumo do aquecimento ambiente é alta. Ao contrário do caso dos frigoríficos e máquinas, os consumos dos aparelhos de aquecimento ambiente na base de dados do *re:dy* não podem ser considerados representativos do consumo total de climatização da casa. Não existem, portanto, observações fiáveis da variável de interesse (consumo parcial da classe de aquecimento ambiente elétrico), o que dificulta a sua previsão através de modelos estatísticos. Então, foi desenvolvido um algoritmo que não requeresse observações do consumo parcial completo para o estimar.

Colocou-se a seguinte teoria: "Os aparelhos de climatização são responsáveis pela maioria da sazonalidade anual do consumo dos clientes, o que permite efetuar previsões sobre o seu consumo observando o consumo global do cliente ao longo de um ano". Para confirmar a teoria proposta, comparou-se a média de consumo dos equipamentos de cada classe nos meses de verão e nos meses de inverno (Tabela 5.1).

Tabela 5.1: Diferenças entre as médias de consumo nos meses verão e nos meses de inverno para cada categoria de equipamentos.

Classe de equipamentos	Verão - Inverno
Águas Quentes Sanitárias	-22,12 kWh
Aquecimento	-18,81 kWh
Arrefecimento	-10,77 kWh
Refrigeração	9,79 kWh
Máquinas	-1,80 kWh
Iluminação	-0,12 kWh
Multimédia	-0,07 kWh
Cozinha	-0,07 kWh
Informática	-0,02 kWh

As maiores diferenças no consumo médio entre os meses das duas estações do ano verificam-se para os aparelhos das seguintes classes:

- Águas Quentes Sanitárias: os aparelhos de aquecimento de água elétricos (como os termoacumuladores) parecem ser os aparelhos que mais impactam singularmente a variação anual do consumo global.
- Aquecimento: aparelhos de aquecimento elétricos, (como termoventiladores)
- Arrefecimento: maioritariamente unidades de ar condicionado (capazes tanto de arrefeci-

mento ambiente como de aquecimento ambiente). A maior utilização nos meses de inverno sugere que a maior parte do consumo se deve a aquecimento ambiente.

- Refrigeração: os aparelhos de refrigeração (como frigoríficos ou arcas) são uma das raras classes que consome mais energia no verão que no inverno.

5.1 Formulação do Algoritmo

Uma vez que, devido ao impacto relevante do aquecimento de águas e da refrigeração, a teoria proposta não se verificou, propôs-se outra formulação para estimar o consumo de aquecimento ambiente.

Esta formulação baseia-se na existência de períodos do ano em que não se espera utilização de aquecimento ambiente, nomeadamente os meses mais quentes. Estes meses podem ser identificados observando a mediana do consumo de todos os aparelhos de aquecimento ambiente para cada um dos últimos doze meses (Tabela 5.2).

Tabela 5.2: Valores das medianas do consumo para os últimos doze meses (com referência no final de junho de 2018).

Mês	Mediana do Consumo (kWh)
Setembro de 2017	0,21
Junho de 2018	0,32
Julho de 2017	0,42
Agosto de 2017	0,43
Outubro de 2017	0,59
Maio de 2018	1,49
Abril de 2018	12,24
Novembro de 2017	12,64
Março de 2018	18,90
Fevereiro de 2018	22,66
Dezembro de 2017	23,54
Janeiro de 2018	24,38

No horizonte temporal da Tabela 5.2 considerou-se que os meses de julho, agosto, setembro e outubro de 2017 e maio e junho de 2018 seriam os meses em que o consumo de aquecimento seriam desprezáveis (meses de referência). O consumo de aquecimento nestes meses é tipicamente baixo, o que os torna em bons meses de referência para a estimação do consumo de aquecimento nos restantes meses. Além disto, na perspetiva do cliente seria descredibilizante para a empresa que esta apresentasse uma estimativa positiva de consumo de aquecimento ambiente nos meses mais quentes quando este consumo não tivesse ocorrido. Em contraste, a apresentação de uma sobre-estimativa de consumo de aquecimento num mês de inverno seria considerada plausível (ainda que errada).

Propõe-se a seguinte decomposição do consumo global de um cliente para um dado mês j:

$$G_j = \mu + p_j + s_j + a_j + n_j \tag{5.1}$$

Em que:

- G_j: "Consumo global no mês j";
- μ: "Consumo base fixo do cliente (não sazonal)";

- p_j: "Total do consumo medido por *plugs* no mês j";
- s_j: "Componente do consumo sazonal não medida por *plugs* e não ligada ao aquecimento ambiente (aquecimento de águas, arrefecimento ambiente e refrigeração);
- a_j: "Componente não medida por *plugs* do consumo de aquecimento ambiente elétrico"
- n_j: "Ruído aleatório";

Salienta-se nesta formulação, que a componente a_j é considerada nula se j for um mês de referência. Sendo r um mês de referência e k um mês de consumo, considera-se as igualdades (5.2) e (5.3).

$$z_k = G_k - s_k - p_k = \mu + a_k + n_k \tag{5.2}$$

$$z_r = G_r - s_r - p_r = \mu + 0 + n_r \tag{5.3}$$

Nesta formulação, a componente z foi designada de consumo remanescente, ou seja, a componente não medida por *plugs* do consumo global excluindo os fatores sazonais não relacionados com o aquecimento ambiente. A sazonalidade da componente z ao longo dos meses deve-se apenas ao consumo de aquecimento ambiente não medido por *plugs* (5.4).

$$z_k - z_r = a_k + \underbrace{(n_k - n_r)}_{\text{Ruído}} \tag{5.4}$$

O consumo de aquecimento ambiente desconhecido é então estimado subtraindo o consumo remanescente dos meses de referência ao consumo remanescente dos meses de utilização (5.5).

$$\hat{a}_k = \hat{z}_k - \hat{z}_r = (G_k - \hat{s}_k - p_k) - (G_r - \hat{s}_r - p_r) \tag{5.5}$$

Este método de estimação levanta um problema: a necessidade de estimar os consumos de aquecimento de águas, ar condicionado/arrefecimento e refrigeração, caso estes não sejam medidos por *plugs*. Decidiu-se então, que se observaria os resultados da aplicação da formulação exposta para clientes para os quais se tem conhecimento de que possuem aparelhos de aquecimento ambiente mas que não possuem arrefecimento ambiente nem aquecimento de águas elétrico. Estes clientes foram identificados pela funcionalidade descrita na secção 3.1.2 (limitação 5). Esta funcionalidade inclui uma opção que permite aos clientes informar se utilizam ar condicionado para aquecimento, arrefecimento ou ambos. Assim, de entre os utilizadores de ar condicionado, apenas se excluiu os clientes que utilizassem arrefecimento ambiente. Além disto, houve o cuidado de excluir clientes utilizadores de bomba de piscina pois, ainda que essa categoria não exista na classificação das *plugs*, espera-se que tenha um impacto bastante significativo na sazonalidade do consumo global. Assim, foram selecionados os 188 clientes que correspondiam a esta descrição de entre todos os clientes EDP *re:dy*.

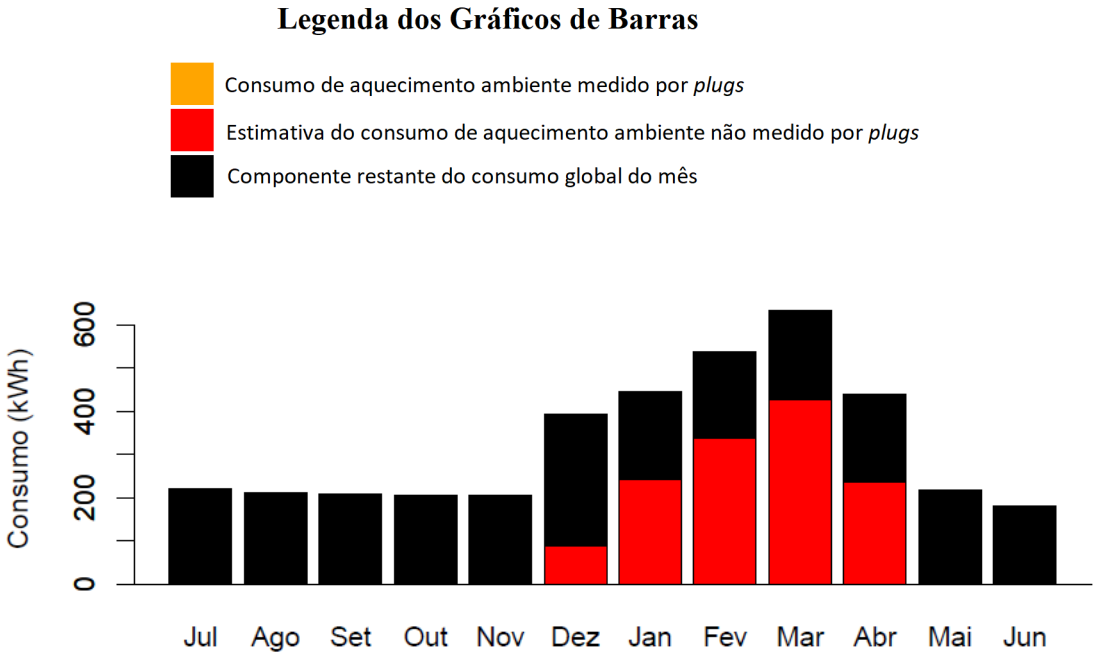
O único fator disruptor restante, a refrigeração, foi estimado através de uma média global do consumo dos frigoríficos da base de dados para cada mês (quando não existiam *plugs* de refrigeração). Esta estimativa é calculada de forma mais computacionalmente rápida do que sofisticada devido ao impacto menos relevante da refrigeração na sazonalidade anual do consumo relativamente às restantes categorias mencionadas (aquecimento ambiente, ar condicionado e aquecimento de águas).

Ao algoritmo apresentado, foram feitas as seguintes alterações antes de ser aplicado aos 188 clientes:

- Uma vez que o conjunto de meses de referência (R) escolhido é constituído por mais que um mês, o cálculo de $\hat{a}_k$ foi efetuado da seguinte forma para cada um dos 12 meses: $\hat{a}_k = \hat{z}_k - \frac{1}{6} \sum_{r \in R} \hat{z}_r$
- Como o objetivo do projeto é alertar o cliente para a possibilidade de excesso de consumo em aquecimento ambiente, foi definido um limite inferior para as estimativas $\hat{a}_k$ para evitar apresentar estimativas quando estas fossem irrelevantes. Considerou-se que não se justificava a notificação do cliente quando o seu consumo mensal em aquecimento elétrico não superasse 15 kWh e 15% do consumo total do cliente. Assim, o valor 0 foi atribuído às estimativas a_k que não superam o seguinte *threshold*: $max\{15; 0.15 \times G_k\} kWh$.

5.2 Resultados

Para ilustrar os resultados obtidos, apresentam-se alguns gráficos de barras que se consideraram representativos dos vários casos. A altura das barras representa o consumo global do cliente no mês respetivo. A área colorida de laranja corresponde aos valores observados de consumo de aquecimento ambiente e a de vermelho corresponde às estimativas de consumo de aquecimento ambiente não medido por *plugs*. O consumo global que se estima não se dever à utilização de aparelhos de aquecimento ambiente está a preto.



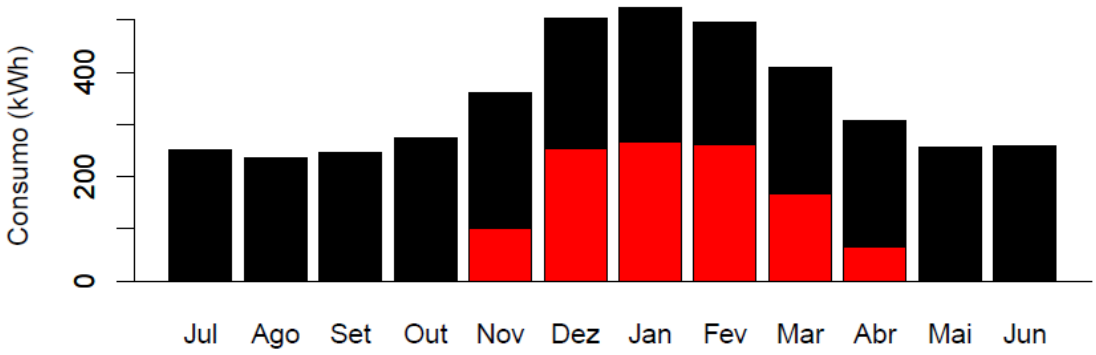


Figura 5.1: Gráficos de barras representativos das estimativas do consumo de aquecimento ambiente e consumos globais mensais dos clientes 6 e 22 (respetivamente).

A grande maioria dos 188 clientes selecionados, não possui *plugs* associadas a aparelhos de aquecimento ambiente. Uma vez que esta análise foi feita durante os meses mais quentes, a falta de *plugs* nesta categoria poderá dever-se à alteração das *plugs* para outros aparelhos (uma vez que poderiam não estar a ser usadas nos meses quentes).

O tipo de resultado mais comum da aplicação do algoritmo é o ilustrado pelos cliente 6 e 22 (Figura 5.1). Estes são os casos em que o consumo nos meses frios se destaca claramente do consumo dos meses quentes por ser bastante mais elevado. Sabendo que estes clientes não possuem aparelhos de aquecimento de águas elétricos, esta curva dever-se-á provavelmente à utilização dos aparelhos de aquecimento ambiente.

Nos raros casos em que existem *plugs* de aquecimento ambiente, desconhece-se se o valor observado de consumo de aquecimento é o total real da categoria. Nestes casos, o algoritmo acrescenta uma estimativa da parte do consumo de aquecimento não medido ao já observado (Figura 5.2).

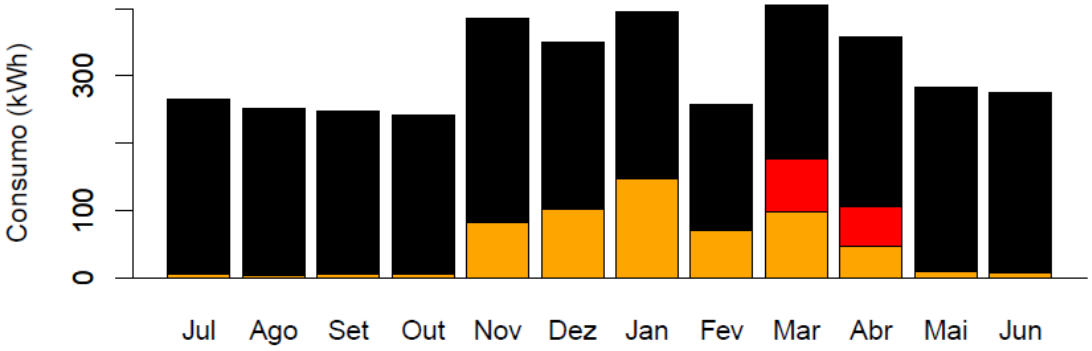


Figura 5.2: Gráfico de barras representativo dos valores observados do consumo de aquecimento, estimativas do consumo de aquecimento ambiente além do observado e consumos globais mensais do cliente 102.

Os clientes com consumo inferior nos meses de inverno em relação aos meses de verão são os casos em que o algoritmo não consegue identificar utilização de aparelhos de aquecimento ambiente. Nestes casos a estimativa a_k original tende a ser negativa para os meses frios e, consequentemente, a ser convertida para zero por estar abaixo do *threshold* positivo (Figura 5.3).

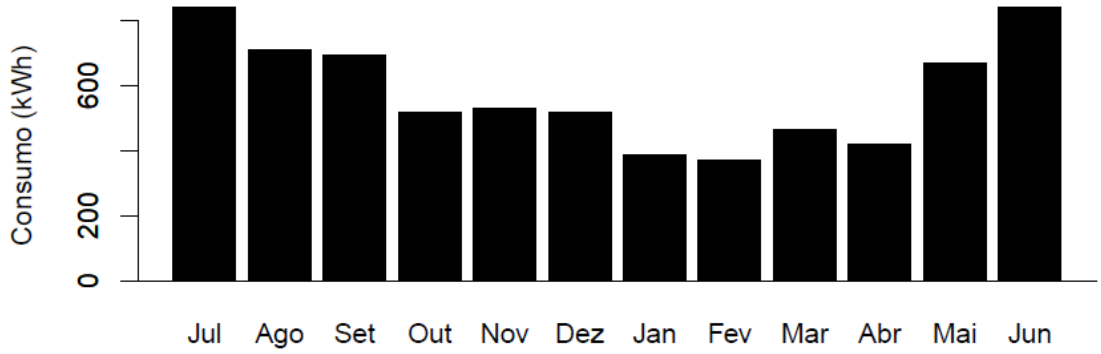


Figura 5.3: Gráfico de barras representativo de estimativas de consumo de aquecimento ambiente e consumo global do cliente 30.

O único cliente para o qual se poderá dizer com alguma certeza que apresenta a totalidade ou a quase totalidade do seu consumo de aquecimento ambiente medido, pela elevada proporção do seu consumo global que pertence à categoria, é o cliente 120. Para este cliente fez-se a experiência de ocultar as medições de aquecimento e obter estimativas para comparar com os valores reais (Figura 5.4).

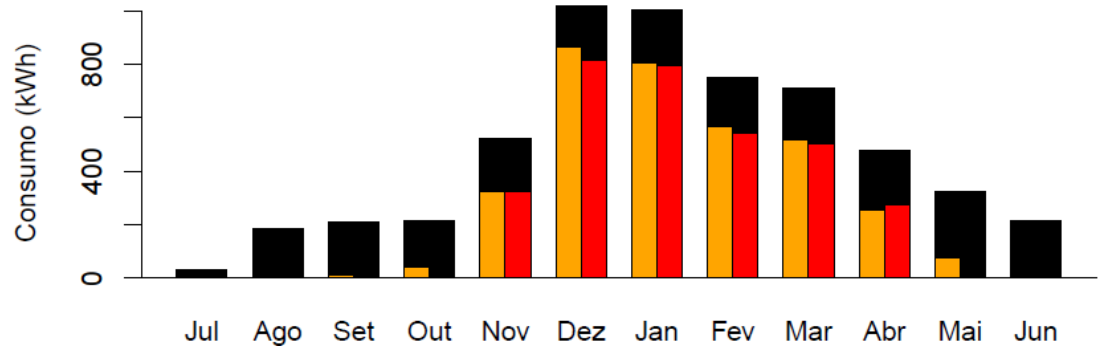


Figura 5.4: Gráfico de barras para efeito de comparação dos valores observados de consumo de aquecimento ambiente (a laranja) e respetivas estimativas (a vermelho) - cliente 120.

Na Figura 5.4 observa-se o bom desempenho do algoritmo neste caso. Ainda que este exemplo não seja suficiente por si só para confirmar o desempenho geral do algoritmo, observa-se que as estimativas apresentadas para os restantes clientes são, pelo menos, plausíveis.

6. Discussão

As precisões máximas (*Energy Accuracy*) do algoritmo de *Ensemble Learning* para os frigoríficos e combinados (entre 55% e 60%) ficaram bastante aquém dos resultados apresentados por Batra et al. [5], estudo em que se obteve uma precisão máxima de 69% utilizando um algoritmo de agrupamento por vizinhos mais próximos. No caso da amostra conjunta de frigoríficos e combinados, a dimensão da amostra (67 clientes) é superior à da experiência de Batra et al. (57 clientes) e a avaliação dos algoritmos foi efetuada através de uma validação cruzada *Leave-One-Out* em ambos os casos. A precisão de 35,76% obtida pelo algoritmo de agrupamento por vizinhos mais próximos no presente estudo poderá dever-se ao facto de os diferentes tipos de frigoríficos não surgirem com a mesma distribuição probabilística em Portugal que nos Estados Unidos. A informação presente nas covariáveis utilizadas no estudo poderá não ser suficiente para a previsão deste tipo de consumo parcial, pelo que a recolha de valores de outras variáveis poderá melhorar o desempenho dos algoritmos. Várias calculadoras de consumo de refrigeração *online* (das quais se destaca a da *Energy Star* [25]) consideram que a data aproximada de compra, a capacidade e o tipo de frigorífico sejam suficiente para determinar com alguma certeza o consumo destes aparelhos. A aplicação para *smartphones* do EDP *re:dy* tem integrada a tecnologia necessária para recolher informação através de questionários ao cliente, pelo que a recolha de valores destas variáveis poderá ser um passo no sentido da melhor precisão na estimação do consumo parcial dos aparelhos de refrigeração. No entanto, a modelação do consumo parcial a partir destas três variáveis não permitiria ao *re:dy* identificar o mau funcionamento de um aparelho caso este estivesse em más condições.

A previsão do consumo das máquinas de lavar constituí um problema mais complicado que a da refrigeração, uma vez que esta secção do consumo global depende bastante do comportamento do cliente. Por ser um consumo relativamente esporádico, é uma classe cujo consumo é, geralmente, eclipsado no consumo global mensal da casa, o que torna difícil a sua explicação através do conjunto de variáveis utilizado que é primariamente baseado em valores de consumo global recolhidos com baixa frequência. O Gemello [5] e a estrutura preditiva utilizada no presente trabalho são exemplos desta situação, pois produzem resultados pouco precisos por se basearem em variáveis não relacionadas com o comportamento dos clientes para desagregar o consumo. Na publicação que apresenta o *DDSC* [4] (um algoritmo de desagregação de consumos de frequência baixa mais sofisticado), observa-se que os resultados da sua aplicação não são consistentemente exatos no caso dos equipamentos informáticos e máquinas, apesar de conseguir identificar alguns períodos de utilização, conhecendo apenas dados de consumo recolhidos de hora a hora. As melhores soluções para o controlo do consumo destes aparelhos continua a ser a sua medição direta (através de *plugs*) ou a utilização de tecnologias com maior frequência de amostragem.

Os resultados da aplicação da estrutura de algoritmos mais precisos no presente trabalho são maioritariamente atribuídos a algoritmos simples como médias ou modelos lineares (como acontece no caso dos frigoríficos). No entanto, se se efetuasse a mesma análise numa amostra de

clientes de dimensão bastante superior, esperaria-se que os modelos estatísticos de aprendizagem automática (redes neuronais, *gradient boosting*) produzissem melhores resultados, a custo de um grande aumento na complexidade computacional. Estes modelos têm a vantagem de conseguir captar a variabilidade das variáveis resposta através de covariáveis fracamente relacionadas com as primeiras, se a dimensão amostral for suficientemente grande (que não é o caso). Embora não tenha sido suficiente para prever os consumos parciais com grande precisão, existe possibilidade de aplicação desta abordagem com maior sucesso no futuro se estiver disponível um conjunto de dados de maior dimensão.

O processo de estimação para o consumo parcial ligado ao aquecimento ambiente produz estimativas aparentemente mais próximas da realidade que a estrutura de algoritmos mais complexa utilizada para os equipamento de refrigeração e máquinas de lavagem. Apesar de não ter sido formalmente testado (comparando os valores reais com os valores preditos), o algoritmo parece ser relativamente adequado para utilização por parte da empresa, pois ainda que não identifique os consumos de aquecimento ambiente quando estes são eclipsados por outras classes de equipamentos, é capaz de os identificar quando estes são realmente acentuados. A implementação deste algoritmo pode ser bastante útil para os clientes *re:dy*, na medida em que quanto maior for o consumo dos aquecimentos mais provável é que seja detetado, notificando o cliente nas situações mais urgentes. Uma extensão deste tipo de processo para estimação do consumo parcial diário (ao invés de apenas mensal) permitiria ao *re:dy* detetar equipamentos que foram deixados por desligar e corrigir a falha antes que resultasse num incremento excessivo no consumo total da habitação. Tal extensão poderia fazer uso da informação disponível sobre o tempo atmosférico, que afeta a utilização desta classe de aparelhos diariamente.

Bibliografia

[1] Gupta, A., Chakravarty, P. (2013). *Impact of energy disaggregation on consumer behavior.* Behavior, Energy, and Climate Change Conference, Sacramento, California.

[2] Hart, G.W. (1992) *Nonintrusive Appliance Load Monitoring.* Proceedings of the IEEE, Vol. 80, No. 12

[3] Kolter, J. Z., Jaakkola, T. (2012). *Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation.* In Journal of Machine Learning Research : Workshop and Conference Proceedings, 22, Pág. 1472-1482.

[4] Kolter, J. Z., Batra, S., Ng, A. (2010). *Energy Disaggregation via Discriminative Sparse Coding.* In Proceedings of the 24th Annual Conference on Neural Information Processing Systems, Pág. 1153-1161, Vancouver, BC, Canadá.

[5] Batra, N., Singh, A., Whitehouse, K. (2016). *Gemello: Creating a Detailed Energy Breakdown from Just the Monthly Electricity Bill.* In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, Nova Iorque, EUA, Pág. 431-440.

[6] Dong, H., Wang, B., Lu, C.T. (2013). *Deep sparse coding based recursive disaggregation model for water conservation.* In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (Pág. 2804–2810): AAAI Press.

[7] Hart, G.W. (1984) *Nonintrusive Appliance Load Data Acquisition Method.*, MIT Energy Laboratory Technical Report

[8] Hastie, T., Tibshirani, R., Friedman, J. (2001). *The Elements of Statistical Learning.* Nova Iorque, EUA: Springer New York Inc..

[9] Batra, N., Singh, A., Whitehouse, K. (2015). *Neighbourhood NILM: A Big-data Approach to Household Energy Disaggregation*

[10] Medium - Human in a Machine World [Online]. *MAE and RMSE – Which Metric is Better?* Disponível em: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

[11] Elite Data Science [Online]. *Overfitting in Machine Learning: What It Is and How to Prevent It.* Disponível em: https://elitedatascience.com/overfitting-in-machine-learning

[12] Jolliffe, I. T. (2002). *Principal Component Analysis. Second ed.* Springer Series in Statistics. Nova Iorque: Springer-Verlag New York.

[13] Cabral, M. S. (2015). Apontamentos da disciplina de Modelo Linear e Extensões. Licenciatura em Estatística Aplicada. Lisboa, FCUL.

[14] Legendre, A. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes.* Firmin Didot, Paris.

[15] Rousseeuw, P. J., Leroy, A. M. (1987) *Robust Regression and Outlier Detection.* Wiley.

[16] Box, G., Cox, D. (1964). *An Analysis of Transformations.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 26, No. 2.

[17] Fix, E. and Hodges, J. (1951). *Discriminatory analysis—nonparametric discrimination: Consistency properties.* Technical Report 21-49-004,4, U.S. Air Force, School of Aviation Medicine, Randolph Field, TX.

[18] Pelillo, M. (2014). *Alhazen and the nearest neighbor rule.* Pattern Recognition Letters, ISSN: 0167-8655, Vol.: 38, Edição: 1, Pág.: 34-37

[19] Gorman, B. (2017) [Online] *A Kaggle Master Explains Gradient Boosting.* Disponível em: http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/

[20] Kaggle: Your Home for Data Science [Online] *Competitions – Kaggle* Disponível em: https://www.kaggle.com/competitions

[21] Apache Hadoop [Online] Disponível em: http://hadoop.apache.org

[22] Feature engineering in data science [Online] Disponível em: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features

[23] Gorman, B. (2016) [Online] *A Kaggler's Guide to Model Stacking in Practice.* Disponível em: http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/

[24] Souza, B., Brito, N., Neves, W., Silva, K., Lima, R., Silva, S. (2004). *Comparison between backpropagation and RPROP algorithms applied to fault classification in transmission lines.* Proceedings 2004 IEEE International Joint Conference on Neural Networks Volume: 4

[25] Energy Star Refrigerator Calculator | ENERGY STAR [Online] Disponível em: https://www.energystar.gov/index.cfm?fuseaction=refrig.calculator

Apêndice

A Interface do algoritmo preditivo (frigoríficos e máquinas)

```
# Input ####
categoria<-2 #2 - refrigeracao, 3 - maquinas
subcategoria<-1
ftriagem="wash whitelist" #lista de clientes que passaram na triagem
whichcomponents<-"best"#"first"# #escolha do conjunto de componentes principais
    ↪  a utilizar


krange<-1:7 #valores a testar para o numero de vizinhos no algoritmo de
    ↪ agrupamento por vizinhos mais proximos
r<- 6#num componentes a reter
trainprop<-1;nreps<-5 #proporcao de observacoes
seed<-0 #se seed=0 nao utiliza seed
nfolds<-5 #numero de folds na validacao cruzada para obtencao das previsoes de
    ↪ primeiro nivel
datnum<-"526" #identificacao da amostra a utilizar: amostra final com 526
    ↪ clientes - "526"
useagreg<-0 #inclusao da dimensao do agregado familiar nas covariaveis
varctrl<-5;nsamp<-"best"
triagem<-1
# (mean,knn,nnet,lm,glm,lts,xgb)
interruptor1<-c (1, 1, 1, 1, 1, 1, 0 )
# (e_mean,e_decision,e_weighting,stack_lm,stack_nnet)
interruptor2<-c (1, 1, 1, 1, 1 )
#------------------------------------------------------------------


# Inicializacoes ####
#execucao do ficheiro Stacking Initializations.R,
#que inicializa as matrizes de dados (JoinMat, data e comp) conforme a
    ↪ configuracao dada na interface
source("Stacking Initializations.R")
#------------------------------------------------------------------


#Algoritmo####
{ tinicial<-Sys.time(); source("Stacking Algorithm 3.R"); tfinal<-Sys.time()}
#------------------------------------------------------------------


print(tfinal-tinicial)
```

B Algoritmo preditivo (frigoríficos e máquinas)

```
# INICIO DO ALGORITMO PREDITIVO ####


JM<-JoinMat #matriz com valores de consumo parcial para cada mes e valores das
```

```
    ↪ 6 componentes principais escolhidas para cada um dos clientes
    ↪ selecionados para a experiencia
D<-data #matriz com uma coluna vazia para previsoes de consumo parcial e
    ↪ valores de todas as componentes principais para os clientes selecionados
    ↪  para a experiencia
COMP<-comp #matriz com os valores das estimativas componentes principais para
    ↪ os 526 clientes
for(repet in 1:reps){
  # Divisao entre conjunto de validacao e conjunto de treino ####
  cat("\n\nrep",repet,"\n")
  aux<-sample(1:length(lin))
  trainset<-sort(head(aux,floor(trainprop*length(aux))))
  testset<-sort(aux[!aux%in%trainset])
  if(trainprop==1){
    trainset<-(1:length(lin))[-repet]
    testset<-repet
  }

  # Selecao das covariaveis a usar na iteracao atual ####
  {
    if(whichcomponents=="first"){
      components<-1:r
    }
    if(whichcomponents=="best"){
      auxm<-JM[trainset,]
      components<- which(colnames(COMP)%in%names(head(sort(colMeans(abs(cor(
          ↪ auxm)[1:12,-c(1:12)])),decreasing = T),r)))
    }

    JoinMat<-JM[,c(1:12,12+components)]
    data<-D[,c(1,1+components)]
    comp<-COMP[,components]

        # Inclusao do numero de pessoas no agregado familiar como covariavel
            ↪ ####
    if(useagreg==1){
      JoinMat<-cbind(JoinMat,coluna_agreg)
      colnames(JoinMat)[1:12]<-paste0("X",1:12)
      colnames(JoinMat)[12+r+1]<-"agreg"
      JoinMat<-as.data.frame(JoinMat)
      data<-cbind(data,coluna_agreg)
      colnames(data)[length(colnames(data))]<-"agreg"
    }
  }

  testnames<-c(testnames,testset)
```

```
train<-data[trainset,]
train <- train[ order(row.names(train)), ]
test<-data[testset,]
test <- test[ order(row.names(test)), ]

# Separacao do conjunto de treino em folds ####
n[1:nfolds]<-length(trainset)%/%nfolds
if(length(trainset)%%nfolds>0){
  n[1:(length(trainset)%%nfolds)]<- n[1:(length(trainset)%%nfolds)]+1
}
folds<-factor(sample(rep(1:nfolds,n[1:nfolds])))
train<-cbind(train,folds)

# Definicao de matrizes de dados para os meta-algoritmos ####
train_meta<-train
test_meta<-test
for(modn in 1:length(lvl1_models)){
  train_meta<-cbind(train_meta,rep(NA,length(trainset)))
  test_meta<-cbind(test_meta,rep(NA,length(testset)))
}
colnames(train_meta)<-c(colnames(train),lvl1_models)
colnames(test_meta)<-c(colnames(test),lvl1_models)
Meta_Train<-Meta_Test<-list()
for(meses in 1:12){
  Meta_Train[[meses]]<-train_meta
  Meta_Test[[meses]]<-test_meta
  Meta_Train[[meses]]$MonCons<-y[trainset,meses]
  Meta_Test[[meses]]$MonCons<-y[testset,meses]

}
t1<-Sys.time()
# Validacao cruzada dentro do conjunto de treino ####
for(ix in 1:(nfolds+1)){
  if(ix<=nfolds){
    cat("fold",ix,">")
    metatrain_foldtest<-which(train_meta$folds==ix)
    metatrain_foldtrain<-which(train_meta$folds!=ix)
    foldtest<-trainset[metatrain_foldtest]
    foldtrain<-trainset[metatrain_foldtrain]
    # Media do conjunto de treino - Validacao cruzada ####
    if(interruptor1[1]==1){
      mean.pred<-colMeans(JoinMat[foldtrain,1:12])
      mean.pred=matrix(rep(mean.pred,length(foldtest)),
                       ncol=12,
                       byrow=T)
      cat("M")
```

```
}

# Vizinhos mais proximos - Validacao cruzada ####
if(interruptor1[2]==1){
  knn.fit<-fit.gemello(matapp,comp,lin[foldtrain],1:7)
  knn.pred<-predict.gemello(knn.fit,matapp,comp,lin[foldtrain],lin[
      ↪ foldtest])
  cat("K")
}

# Rede Neuronal - Validacao cruzada ####
if(interruptor1[3]==1){

  data<-JoinMat
  maxs <- apply(data[foldtrain,], 2, max)
  mins <- apply(data[foldtrain], 2, min)
  scaled <- as.data.frame(scale(data, center = mins, scale = maxs - mins)
      ↪ )

  nntrain<-scaled[foldtrain,]
  nntest<-scaled[foldtest,]




  nnnames <- names(nntrain)
  f <- as.formula(paste("X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12␣~", paste
      ↪ (nnnames[!nnnames %in%c("X1","X2","X3","X4","X5","X6","X7","X8","
      ↪ X9","X10","X11","X12") ], collapse = "␣+␣")))
  nn <- neuralnet(f,data=nntrain,hidden=c(5,3),act.fct="logistic",linear.
      ↪ output = F,stepmax=1e6,threshold=0.01)
  pr.nn <- compute(nn,nntest[,-(1:12)])
  center<-mins[1:12]
  scale<-((maxs-mins)[1:12])
  nnet.pred<-pr.nn$net.result
  nnet.pred<-as.data.frame(t(t(nnet.pred) * scale)) #unscale
  nnet.pred<-as.data.frame(t(t(nnet.pred) + center)) #uncenter
  cat("N")
}

# Modelo Linear - Validacao cruzada ####
if(interruptor1[4]==1){
  lmjoinmat<-JoinMat
  lmjoinmat[,1:12]<-lmjoinmat[,1:12]+1e-10
  lm.fit <- lm(cbind(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12) ~ ., data=
      ↪ lmjoinmat[foldtrain,])
```

```
  bc<-boxcox(lm.fit,plotit=F)
  lambda<-boxcoxlambda(bc)

  if(lambda!=0){lmjoinmat[,1:12]<-(lmjoinmat[,1:12]^lambda-1)/lambda}
  if(lambda==0){lmjoinmat[,1:12]<-log(lmjoinmat[,1:12])}
  lm.fit.boxcox<-lm(cbind(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12) ~ .,
      ↪ data=lmjoinmat[foldtrain,])
  lm.pred <- predict(lm.fit.boxcox,newdata=JoinMat[foldtest,])
  if(lambda!=0){
    lm.pred<-(lm.pred*lambda+1)
    lm.pred[which(lm.pred<0)]<-0
    lm.pred<-lm.pred^(1/lambda)
  }
  if(lambda==0){lm.pred<-exp(lm.pred)}
  lm.pred[lm.pred<0]<-0
  cat("L")
}


# Modelo Linear Generalizado - Validacao cruzada ####
if(interruptor1[5]==1){
  tc<-NA
  noise<-0.001
  while(!is.null(tc)){
    tc<-tryCatch(
      {
        NoiseJoinMatTrain<-JoinMat[foldtrain,]+matrix(runif(dim(JoinMat[
            ↪ foldtrain,])[1]*dim(JoinMat[foldtrain,])[2],0,noise),nrow=
            ↪ dim(JoinMat[foldtrain,])[1],byrow = T)
        glm.fit <- list()
        dvnames <- c("X1", "X2", "X3", "X4", "X5" ,"X6" ,"X7" ,"X8" ,"X9"
            ↪ ,"X10" ,"X11" ,"X12")
        ivnames <- paste0("PC",components[1],"+PC",components[2],"+PC",
            ↪ components[3])
        glm.pred<-matrix(NA,ncol=12,nrow=length(foldtest))
        for (dv in 1:length(dvnames)){
          form <- formula(paste(dvnames[dv], "~", ivnames))
          glm.fit[[dv]] <- glm(form, data=NoiseJoinMatTrain, family=Gamma(
              ↪ link="log"))
          glm.pred[,dv]<-exp(predict.glm(glm.fit[[dv]],newdata=JoinMat[
              ↪ foldtest,]))
        }
      },error = function(err) {
        cat("")
        return(NA)
      },finally = { 1
```

```
      }
    )
    noise<-ifelse(noise<1,noise*2,noise)
  }
  noise<-noise/2;if(noise>0.001){cat("\nnoise␣limit␣increased␣to",noise,"
      ↪ \n")}
  cat("G")
}


# Modelo de Regressao Robusta - Validacao cruzada ####
if(interruptor1[6]==1){
  {
    lts.fit <- list()
    dvnames <- c("X1", "X2", "X3", "X4", "X5" ,"X6" ,"X7" ,"X8" ,"X9" ,"
        ↪ X10" ,"X11" ,"X12")
    ivnames <- capture.output(cat(colnames(JoinMat)[-(1:12)],sep="+"))
    lts.pred<-matrix(NA,ncol=12,nrow=length(foldtest))
    for (dv in 1:length(dvnames)){
      form <- formula(paste(dvnames[dv], "~", ivnames))
      matlts<-c()
      for(ixlts in 1:varctrl){
        matlts<-rbind(matlts,lqs(form, data=JoinMat[foldtrain,],method="
            ↪ lts",na.action = na.omit,nsamp=nsamp)$coefficients)
      }
      lts.fit[[dv]] <- lqs(form, data=JoinMat[foldtrain,],method="lts",na.
          ↪ action = na.omit)
      lts.fit[[dv]]$coefficients<-colMeans(matlts)
      lts.pred[,dv]<-predict(lts.fit[[dv]],newdata=JoinMat[foldtest,])
    }
  }
  lts.pred[lts.pred<0]<-0
  cat("R")
}


# Gradient Boosting - Validacao cruzada ####
if(interruptor1[7]==1){
  sinkall();sink("Scheduler␣Results/xgb_output.txt")
  xgb_grid_1 = expand.grid(nrounds = c(1000,3000) ,
                           eta = c(0.001, 0.0001),
                           lambda = 1,
                           alpha = 0)
  xgb_trcontrol_1 = trainControl(method = "cv",
                                 number = 5,
                                 verboseIter = TRUE,
```

```
                                returnData = FALSE,
                                returnResamp = "all",
                                allowParallel = TRUE)

    xgb.pred<-matrix(NA,ncol=12,nrow=length(foldtest))
    for (dv in 1:length(dvnames)){
      xgb_train_1 = train(x=JoinMat[foldtrain,13:(12+r+useagreg)],y=JoinMat[
          ↪ foldtrain,dv],trControl = xgb_trcontrol_1,
                          tuneGrid = xgb_grid_1,
                          method = "xgbLinear",
                          max.depth = 5)
      xgb.pred[,dv]<-predict(xgb_train_1,newdata=JoinMat[foldtest,13:(12+r+
          ↪ useagreg)])
    }
    sinkall();sink("Scheduler␣Results/console_output.txt",append=T)
    cat("X")
  }


  # Estimativas de validacao cruzada para o conjunto de treino ####
  for(mon in 1:12){
    if(interruptor1[1]==1) {Meta_Train[[mon]]$mean[metatrain_foldtest]<-(
        ↪ mean.pred)[,mon]}
    if(interruptor1[2]==1) {Meta_Train[[mon]]$knn[metatrain_foldtest]<-(knn
        ↪ .pred)[,mon]}
    if(interruptor1[3]==1) {Meta_Train[[mon]]$nnet[metatrain_foldtest]<-(
        ↪ nnet.pred)[,mon]}
    if(interruptor1[4]==1) {Meta_Train[[mon]]$lm[metatrain_foldtest]<-(lm.
        ↪ pred)[,mon]}
    if(interruptor1[5]==1) {Meta_Train[[mon]]$glm[metatrain_foldtest]<-(glm
        ↪ .pred)[,mon]}
    if(interruptor1[6]==1) {Meta_Train[[mon]]$lts[metatrain_foldtest]<-(lts
        ↪ .pred)[,mon]}
    if(interruptor1[7]==1) {Meta_Train[[mon]]$xgb[metatrain_foldtest]<-(xgb
        ↪ .pred)[,mon]}
  }
  cat("\n")

}

if(ix==nfolds+1){
  # Previsao dos algoritmos de primeiro nivel para o conjunto de validacao
      ↪ ####
  cat("predict>")
  # Media do conjunto de treino - Primeiro nivel ####
  if(interruptor1[1]==1){
```

```
  mean.pred<-colMeans(JoinMat[trainset,1:12])
  mean.pred=matrix(rep(mean.pred,length(testset)),
                   ncol=12,
                   byrow=T)
  cat("M")
}

# Vizinhos mais proximos - Primeiro nivel####
if(interruptor1[2]==1){
  knn.fit<-fit.gemello(matapp,comp,lin[trainset],1:7)
  knn.pred<-predict.gemello(knn.fit,matapp,comp,lin[trainset],lin[testset
      ↪ ])
  cat("K")
}

# Rede neuronal - Primeiro nivel####
if(interruptor1[3]==1){

  data<-JoinMat
  maxs <- apply(data[trainset,], 2, max)
  mins <- apply(data[trainset,], 2, min)
  scaled <- as.data.frame(scale(data, center = mins, scale = maxs - mins)
      ↪ )

  nntrain<-scaled[trainset,]
  nntest<-scaled[testset,]




  nnnames <- names(nntrain)
  f <- as.formula(paste("X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12 ~", paste
      ↪ (nnnames[!nnnames %in% c("X1","X2","X3","X4","X5","X6","X7","X8","
      ↪ X9","X10","X11","X12") ], collapse = " + ")))
  nn <- neuralnet(f,data=nntrain,hidden=c(5,3),act.fct="logistic",linear.
      ↪ output = F,stepmax=2e6,threshold=0.01)
  pr.nn <- compute(nn,nntest[,-(1:12)])
  center<-mins[1:12]
  scale<-((maxs-mins)[1:12])
  nnet.pred<-pr.nn$net.result
  nnet.pred<-as.data.frame(t(t(nnet.pred) * scale)) #unscale
  nnet.pred<-as.data.frame(t(t(nnet.pred) + center)) #uncenter
  cat("N")
```

```
}

# Modelo linear - Primeiro nivel ####
if(interruptor1[4]==1){
  lmjoinmat<-JoinMat
  lmjoinmat[,1:12]<-lmjoinmat[,1:12]+1e-10
  lm.fit<-lm(cbind(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12) ~ ., data=
      ↪ lmjoinmat[trainset,])
  bc<-boxcox(lm.fit,plotit=F)
  lambda<-boxcoxlambda(bc)
  if(lambda!=0){lmjoinmat[,1:12]<-(lmjoinmat[,1:12]^lambda-1)/lambda}
  if(lambda==0){lmjoinmat[,1:12]<-log(lmjoinmat[,1:12])}
  lm.fit.boxcox<-lm(cbind(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12) ~ .,
      ↪ data=lmjoinmat[trainset,])
  lm.pred <- predict(lm.fit.boxcox,newdata=lmjoinmat[testset,])
  if(lambda!=0){lm.pred<-(lm.pred*lambda+1);lm.pred[lm.pred<0]<-0;lm.pred
      ↪ <-lm.pred^(1/lambda)}
  if(lambda==0){lm.pred<-exp(lm.pred)}

  cat("L")
}

# Modelo linear generalizado - Primeiro nivel ####
if(interruptor1[5]==1){
  tc<-NA
  noise<-0.001
  while(!is.null(tc)){
    tc<-tryCatch(
      {#GLM
        NoiseJoinMatTrain<-JoinMat[trainset,]+matrix(runif(dim(JoinMat[
            ↪ trainset,])[1]*dim(JoinMat[trainset,])[2],0,noise),nrow=dim(
            ↪ JoinMat[trainset,])[1],byrow = T)
        glm.fit <- list()
        dvnames <- c("X1", "X2", "X3", "X4", "X5" ,"X6" ,"X7" ,"X8" ,"X9"
            ↪ ,"X10" ,"X11" ,"X12")
        ivnames <- paste0("PC",components[1],"+PC",components[2],"+PC",
            ↪ components[3]) ## for some value of n
        glm.pred<-matrix(NA,ncol=12,nrow=length(testset))
        for (dv in 1:length(dvnames)){
          form <- formula(paste(dvnames[dv], "~", ivnames))
          glm.fit[[dv]] <- glm(form, data=NoiseJoinMatTrain, family=Gamma(
              ↪ link="log"))
          glm.pred[,dv]<-exp(predict.glm(glm.fit[[dv]],newdata=JoinMat[
              ↪ testset,]))
        }
      },error = function(err) {
```

```
        cat("")
        return(NA)
      },finally = { 1
      }
    )
    noise<-noise*2

  }
  noise<-noise/2;if(noise>0.001){cat("\nnoise limit increased to",noise,"
      ↪ \n")}
  cat("G")
}

# Modelo de regressao robusta - Primeiro nivel ####
if(interruptor1[6]==1){
  {
    lts.fit <- list()
    dvnames <- c("X1", "X2", "X3", "X4", "X5" ,"X6" ,"X7" ,"X8" ,"X9" ,"
        ↪ X10" ,"X11" ,"X12")
    ivnames <- capture.output(cat(colnames(JoinMat)[-(1:12)],sep="+"))
    lts.pred<-matrix(NA,ncol=12,nrow=length(testset))
    for (dv in 1:length(dvnames)){
      form <- formula(paste(dvnames[dv], "~", ivnames))
      matlts<-c()
      for(ixlts in 1:varctrl){
        matlts<-rbind(matlts,lqs(form, data=JoinMat[trainset,],method="lts
            ↪ ",nsamp=nsamp)$coefficients)
      }
      lts.fit[[dv]] <- lqs(form, data=JoinMat[trainset,],method="lts")
      lts.fit[[dv]]$coefficients<-colMeans(matlts)
      lts.pred[,dv]<-predict(lts.fit[[dv]],newdata=JoinMat[testset,])
    }
  }
  lts.pred[lts.pred<0]<-0

  cat("R")
}

# Gradient Boosting - Primeiro nivel ####
if(interruptor1[7]==1){
  sinkall();sink("Scheduler Results/xgb_output.txt")

  xgb_grid_1 = expand.grid(nrounds = c(1000,3000) ,
                           eta = c( 0.001, 0.0001),
                           lambda = 1,
                           alpha = 0)
```

```
      xgb_trcontrol_1 = trainControl(method = "cv",
                                     number = 5,
                                     verboseIter = TRUE,
                                     returnData = FALSE,
                                     returnResamp = "all",
                                     allowParallel = TRUE)

      xgb.pred<-matrix(NA,ncol=12,nrow=length(testset))
      for (dv in 1:length(dvnames)){
        xgb_train_1 = train(x=JoinMat[trainset,13:(12+r+useagreg)],y=JoinMat[
            ↪ trainset,dv],trControl = xgb_trcontrol_1,
                            tuneGrid = xgb_grid_1,
                            method = "xgbLinear",
                            max.depth = 5)
        xgb.pred[,dv]<-predict(xgb_train_1,newdata=JoinMat[testset,13:(12+r+
            ↪ useagreg)])
      }
      # closeAllConnections()
      sinkall();sink("Scheduler␣Results/console_output.txt",append=T)

      cat("X")
    }


    # Previsoes dos algoritmos de primeiro nivel para o conjunto de teste
        ↪ ####
    for(mon in 1:12){
      if(interruptor1[1]==1){Meta_Test[[mon]]$mean<-(mean.pred)[,mon]}
      if(interruptor1[2]==1){Meta_Test[[mon]]$knn<-(knn.pred)[,mon]}
      if(interruptor1[3]==1){Meta_Test[[mon]]$nnet<-(nnet.pred)[,mon]}
      if(interruptor1[4]==1){Meta_Test[[mon]]$lm<-(lm.pred)[,mon]}
      if(interruptor1[5]==1){Meta_Test[[mon]]$glm<-(glm.pred)[,mon]}
      if(interruptor1[6]==1){Meta_Test[[mon]]$lts<-(lts.pred)[,mon]}
      if(interruptor1[7]==1){Meta_Test[[mon]]$xgb<-(xgb.pred)[,mon]}
    }
    cat("\n")
  }
  if(reps==1) {Progress(ix,nfolds+1,t1)}
}



# NIVEL 2 ####
for(mon in 1:12){

  cat("\n")
```

```
train_meta<-Meta_Train[[mon]]
test_meta<-Meta_Test[[mon]]

train_meta<-train_meta[,-(length(train[1,]))]
# if(trainprop==1){f = file();sink(file=f)} ## silence upcoming output
    ↪ using anonymous file connection
predmat<-test_meta[,(r+1+1+useagreg):(r+1+useagreg+length(lvl1_models))]
ensemble_pred<-predmat

# Media das estimativas ####
if(interruptor2[1]==1)
{#Mean
  EMean.pred<-rowMeans(predmat)
  ensemble_pred<-cbind(ensemble_pred,EMean.pred)
  # closeAllConnections()
  cat("M")
}

# Media ponderada e decisao de estimativas ####
errortrain<-cbind(abs(train_meta$MonCons-train_meta$mean),abs(train_meta$
    ↪ MonCons-train_meta$knn),abs(train_meta$MonCons-train_meta$nnet),abs(
    ↪ train_meta$MonCons-train_meta$lm),abs(train_meta$MonCons-train_meta$
    ↪ glm),abs(train_meta$MonCons-train_meta$lts),abs(train_meta$MonCons-
    ↪ train_meta$xgb))
bestmethod<-factor(apply(errortrain,1,which.min) )#,levels=1:(dim(
    ↪ errortrain)[2]))
trainrpart<-data.frame(bestmethod=bestmethod,train_meta[,-1])
weightingmod<-randomForest(formula=as.factor(bestmethod)~.,data=trainrpart,
    ↪ ntree =2000)
testrpart<-data.frame(bestmethod=rep(NA,dim(test_meta)[1]),test_meta[,-1])
predwei<-matrix(0,nrow=length(lvl1_models),ncol=dim(testrpart)[1])
bmoccur<-sort(unique(bestmethod))
predwei[bmoccur,]<-t(predict(weightingmod,newdata=testrpart,type = c("prob"
    ↪ ),na.action=na.pass))
preddec<-apply(predwei,2,probtobin)
if(interruptor2[2]==1){
  Deci.pred<-rowSums(predmat*t(preddec))
  ensemble_pred<-cbind(ensemble_pred,Deci.pred)
  cat("D")
}
if(interruptor2[3]==1){
  Weig.pred<-rowSums(predmat*t(predwei))
  ensemble_pred<-cbind(ensemble_pred,Weig.pred)
  cat("W")
}
```

```
# if(trainprop<1){f<-file();sink(f)}

# Modelo linear - segundo nivel ####
if(interruptor2[4]==1)
{#LMBOXCOX
  lmtrain<-train_meta
  lmtest<-test_meta
  lmtest$MonCons<-lmtest$MonCons+1e-10
  lmtrain$MonCons<-lmtrain$MonCons+1e-10
  stack.fit<-lm(MonCons~.,data=lmtrain)
  bc<-boxcox(stack.fit,plotit=F)
  lambda<-boxcoxlambda(bc)
  if(lambda!=0){lmtrain$MonCons<-(lmtrain$MonCons^lambda-1)/lambda}
  if(lambda==0){lmtrain$MonCons<-log(lmtrain$MonCons)}
  stack.fit.boxcox<-lm(MonCons~.,data=lmtrain)
  lm.pred <- predict(stack.fit.boxcox,newdata=lmtest)
  if(lambda!=0){lm.pred<-(lm.pred*lambda+1);lm.pred[lm.pred<0]<-0;lm.pred<-
      ↪ lm.pred^(1/lambda)}
  if(lambda==0){lm.pred<-exp(lm.pred)}
  Stacklm.pred<-lm.pred
  ensemble_pred<-cbind(ensemble_pred,Stacklm.pred)
  # closeAllConnections()
  cat("L")
}

# Rede neuronal - segundo nivel ####
if(interruptor2[5]==1)
{#NN
  data<-rbind(train_meta,test_meta)
  stacknntrain<-1:dim(train_meta)[1]
  stacknntest<-(dim(train_meta)[1]+1):dim(data)[1]
  maxs <- apply(data[stacknntrain,], 2, max)
  mins <- apply(data[stacknntrain,], 2, min)
  scaled <- as.data.frame(scale(data, center = mins, scale = maxs - mins))

  nntrain<-scaled[stacknntrain,]
  nntest<-scaled[stacknntest,]




  nnnames <- names(nntrain)
  f <- as.formula(paste("MonCons ~", paste(nnnames[!nnnames %in% c("MonCons"
      ↪ ,"X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11","X12") ],
      ↪  collapse = " + ")))
```

```
      nn <- neuralnet(f,data=nntrain,hidden=c(5,3),act.fct="logistic",linear.
          ↪ output = F,stepmax=2e6,threshold=0.01)

      pr.nn <- compute(nn,nntest[,-(1)])
      center<-mins[1]
      scale<-((maxs-mins)[1])
      stacknn.pred<-pr.nn$net.result
      stacknn.pred<-as.data.frame(t(t(stacknn.pred) * scale)) #unscale
      stacknn.pred<-as.data.frame(t(t(stacknn.pred) + center)) #uncenter
      ensemble_pred<-cbind(ensemble_pred,stacknn.pred)

      cat("N")


    }


     closeAllConnections()
    # Medidas de Diagnostico ####
    A<-maccur(test_meta$MonCons,EMean.pred)
    cat("**␣",monthnames[mon],"␣",round(A,2),"%␣**",sep="")
    repAccuracy<-c(repAccuracy,A)
    GT_ARRAY[,mon,(((repet-1)*ntest+1):(repet*ntest))]<-matrix(rep(JoinMat[
        ↪ testset,mon],length(modelnames)),ncol=length(JoinMat[testset,mon]),
        ↪ byrow=T)
    ERROR_ARRAY[,mon,(((repet-1)*ntest+1):(repet*ntest))]<-
      GT_ARRAY[,mon,(((repet-1)*ntest+1):(repet*ntest))]-t(ensemble_pred)



  }

  if(reps>1) {Progress(repet,reps,tinicial)}


}
{
  ACCURACY_ARRAY[ERROR_ARRAY!=0]<-1-abs(ERROR_ARRAY[ERROR_ARRAY!=0]/GT_ARRAY[
      ↪ ERROR_ARRAY!=0])
  ACCURACY_ARRAY[ERROR_ARRAY==0]<-1-abs(ERROR_ARRAY[ERROR_ARRAY==0])
  ACCURACY_ARRAY[ACCURACY_ARRAY<0]<-0
  ACCURACY_ARRAY<-ACCURACY_ARRAY*100
  dimnames(ACCURACY_ARRAY)<-dimnames(ERROR_ARRAY)<-list(modelnames,monthnames,
      ↪ testnames)
}
```

```
# FIM DO ALGORITMO PREDITIVO ####


PRED_ARRAY<-GT_ARRAY-ERROR_ARRAY



# Resultados ####

#tabela com valores das medidas de desempenho
MAE<-apply(ERROR_ARRAY,c(1),mae)
RMSE<-apply(ERROR_ARRAY,c(1),rmse)
BATRA<-apply(ACCURACY_ARRAY,c(1),mean)
Stacking_Result_Table<-cbind(RMSE,MAE,BATRA)
View(Stacking_Result_Table)
```